University of Coimbra
Portugal

http://computationalcreativity.net/iccc20/

First published 2020

TITLE: Proceedings of the 11th International Conference on Computational Creativity

EDITORS: F. Amílcar Cardoso, Penousal Machado, Tony Veale and João Miguel Cunha

ISBN: 978-989-54160-2-8

Published by the Association for Computational Creativity (ACC)

Copyright notice: all contents of these proceedings by ACC, the Association for Computational Creativity, published under a Creative Commons Attribution (CC BY) license, which allows unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited (https://creativecommons.org/licenses/by/4.0/).
Preface

Welcome to the proceedings of the 11th International Conference on Computational Creativity, or ICCC’20 (see http://computationalcreativity.net/iccc2020/). This year has been a year of firsts for many conferences and academic events, and ICCC is no different. Although our eleventh conference was planned as a physical meeting of researchers in Coimbra, Portugal, from June 29th to July 3rd, the response to the COVID pandemic has necessitated both a move in the calendar (September 7-11) and a move online.

“Computational Creativity” is a term that has grown in usage and acceptance in a diversity of technical areas, not least since its recent adoption by industry. Once upon a time it was a term that was the exclusive preserve of our growing band of “CC” researchers, but now we see it used almost as often in corporate press-releases and academic job postings. Although this has broadened the term from its core meaning, our community still defines it, and the field that it denotes, as the computational nexus that unites art, science, philosophy and, to an increasing extent, cognitive and social psychology, in the understanding of generative systems that exhibit responsibilities or behaviours which unbiased critics might label as “creative.” The social relevance of our research field has never been greater, especially now that AI more generally has made the leap from the popular imagination to practical reality.

This 11th iteration of the ICCC event is also the 10th anniversary of the very first meeting of our CC community as a conference in its own right, rather than as a co-located workshop. That very first conference was held in Lisbon in 2010, and so it is with a sneaking regard for symmetry that the conference is once again back in Portugal a decade later. Despite the growing popularity of the term, the ICCC conference remains the only scientific conference that focuses wholly on computational creativity in all its sub-forms and manifestations.

These proceedings contain a peer-reviewed selection of the 79 full papers that have been submitted for consideration this year, across the following categories: technical papers that pose and address hypotheses about aspects of creativity in computational systems; system papers that describe the building or deployment of a creative system; resource papers that describe the creation of a reusable resource on which other systems can build; study papers that present studies in creativity, perhaps appealing to broader areas of AI and Computer Science or to fields such as psychology or philosophy; application papers that position CC systems in a cultural milieu such as an art exhibition, a concert, a game jam, a poetry or book reading, a cookery demonstration, or some other public-facing outreach event; and position papers that articulate a point-of-view on some aspect of the philosophy, theory, practice or culture of CC research.

Each submission was reviewed in the first instance by the main program committee. A metareview was then conducted by senior committee members and program chairs, who debated the relative merits of each submission. Papers were accepted based on quality, academic rigour and relevance to one of the paper categories outlined above. This process has resulted in a diverse program that reflects the changing trends in artificial intelligence and the state of the art in computational creativity research. In all, the review committee accepted 29 full papers for oral presentation and 14 full papers for poster presentation. A subsequent call for short papers later received 67 submissions across the following 11 categories: Nuggets and Gems, papers that are succinct enough, or early enough, to warrant a shorter format; System Demonstrations, for the Show-and-Tell session; Debate Sparks, for provocations that get the community talking; CC Translations, for presenting relevant work from other fields; CC Bridges, for papers that foster interdisciplinarity by introducing ideas from beyond our conventional horizons; Late Breaking Results, for up-to-the-minute work that
missed the earlier deadline; *Pilot Studies*, for initial forays into a new research topic; *Grand Challenges*, for task proposals that aim to bring the CC community together in a collaborative effort; *Meta-Perspectives*, for papers that reflect on how the CC community might do things differently and better; *Field Reports*, on taking CC research into the field; and *Event Reports* that describe the experience of organizing another CC-themed event.

Several of these paper categories are new to the conference this year, and result from the efforts of our community’s new *CC Task Force*. This is a group of young and enthusiastic researchers that works on behalf of the community to ensure that our conferences and related events (tutorials, workshops, web-site, and other means of outreach) reflect the voices of all of the community, as well as the best practices of academia more generally.

All short paper submissions were reviewed by the program committee, and papers were again accepted based on quality, academic rigour and relevance to one or more of the conference’s short paper categories. The committee accepted 12 papers for short oral presentations, 19 for poster presentation and 10 for a Show & Tell demonstration.

To summarize, then, we received 143 submissions in all, and from these, 84 were accepted for publication in the current proceedings.

The scientific program for ICCC’20 was organised in two stages. In the first, pre-recorded videos for all presentations, including keynote talks, posters and system demonstrations, were made available to the participants one week before the start of the conference. The second stage comprised a series of thematic live sessions across two days, in which the authors of long and short presentations could interact with participants. In addition, two further live sessions were organised with the Keynote Speakers Emilia Goméz (of the European Commission’s Joint Research Centre, and the Music Technology Group, Universitat Pompeu Fabra) and Simon Lucas (of Queen Mary University of London). In parallel, two special sessions, one for Posters and another for Show & Tell, allowed authors of these posters and demos to organise live discussions with participants.

ICCC’20 also continued the tradition of hosting several co-located events, including:

- A Doctoral Consortium for up-and-coming CC researchers;
- Three thematic workshops: on *Casual Creators*; on *Future of Co-Creative Systems*; and on *Knowledge-Based Systems in Computational Design*;
- And, for the first time, an interactive, hands-on workshop: *Introduction to Generative Drawing with pencils, paper, C++ and openFrameworks*;
- Also for the first time, a *Digital Sound Art* event, comprising a virtual exhibition with live sessions for exhibitors and conference participants.
- Three tutorials: *A Deep Dive into Latent Space: Image Generation and Manipulation with StyleGAN2*; *Building Generative Art Tools*; and *Quantum Algorithms for Artistic Experiences*.

As in past years, ICCC’20 also made an award for Best Paper and for Best Student Paper as chosen from this year’s submissions.

**Acknowledgements**
We wish to express our thanks to our sponsors, the Department of Informatics Engineering and the Centre for Informatics and Systems, both at the University of Coimbra. We are also grateful to the program committee and to the senior program committee for their diligence and dedication in carefully reviewing all conference submissions. We also thank all of those involved in the organisation of ICCC’20, including the Local Chair, the Workshop Chairs, the Tutorials Chair, the Doctoral Consortium Chair, the Digital Sound Art Chair, the Proceedings Chair and the ACC steering committee. The CC Task Force has, in its first year of operation, also been a credit to the community and the conference. Finally, the CC community as a whole is also deserving of our gratitude and praise. It is your enthusiasm and energy that has enabled the conference to flourish and grow into what it is today, in its 10th anniversary year: a vibrant meeting place for ideas and possibilities that change the way we think about creativity in both humans and machines.

In Memoriam
Sadly, the 2020 conference cycle was also an occasion of unwelcome news. Robert Keller, a staunch supporter of the CC community and a steadfast and affable presence at the annual conference, passed away shortly after the conclusion of this year’s event. Bob has been a mainstay of our community, as a researcher, collaborator, mentor and reviewer, and he will be missed by all those whose lives he touched. We are reassured to know that his work and his memory will live on.
Program Committee

Senior PC Members

Anna Jordanous - University of Kent
Ashok Goel - Georgia Institute of Technology
Carlos León - Universidad Complutense de Madrid
Dan Ventura - Brigham Young University
Daniel Brown - University of Waterloo
Francois Pachet - CSL Sony Paris
Geraint Wiggins - Vrije Universiteit Brussel / Queen Mary University of London
Hannu Toivonen - University of Helsinki
Julian Togelius - New York University
Kazjon Grace - University of Sydney
Mary Lou Maher - University of North Carolina at Charlotte
Nada Lavrač - Jožef Stefan Institute
Pablo Gervás - Universidad Complutense de Madrid
Rob Saunders - Universiteit Leiden
Simon Colton - Monash University

PC Members

Alan Tapscott - Universidad Complutense de Madrid
Allison Parrish - New York University
Amy K. Hoover - New Jersey Institute of Technology
Ana Rodrigues - University of Coimbra
Ana-Maria Olteteanu - Human-Centered Computing, Freie Universität Berlin
Andrés Gómez de Silva Garza - Instituto Tecnológico Autónomo de México
Anna Kantosalo - Aalto University
Anne Sullivan - Georgia Institute of Technology
Antonios Liapis - Institute of Digital Games, University of Malta
Arne Eigenfeldt - Simon Fraser University
Björn Gambäck - Norwegian University of Science and Technology
Bob Sturm - KTH Stockholm
Brian Magerko - Georgia Institute of Technology
Catherine Havasi - Massachusetts Institute of Technology
Colin Johnson - University of Nottingham
David C Brown - CS Dept. WPI
Contents

Understanding and Strengthening the Computational Creativity Community: A Report From The Computational Creativity Task Force
João M. Cunha, Sarah Harmon, Christian Guckelsberger, Anna Kantosalo, Paul M. Bodily and Kazjon Grace ............................... 1

1 Co-Creativity and Support ................................. 8

1.1 Shaping the Narrative Arc: Information-Theoretic Collaborative Dialogue
Kory Wallace Matheuson, Pablo Samuel Castro, Colin Cherry, George Foster, Marc G. Bellemare ........................................ 9

1.2 Five C’s for Human–Computer Co-Creativity —An Update on Classical Creativity Perspectives
Anna Kantosalo and Tapio Takala ............................. 17

1.3 A Speculative Exploration of the Role of Dialogue in Human–Computer Co-creation
Oliver Bown, Kazjon Grace, Liam Bray and Dan Ventura .............. 25

1.4 Co-Creative Songwriting for Bereavement Support
Lee Cheatley, Margareta Ackerman, Alison Pease and Wendy Moncur ................................................................. 33

1.5 What Happens When a Computer Joins the Group?
Paul Bodily and Dan Ventura ....................................... 41

1.6 Engendering co-creative experiences through agent parametric control
Prashanth Thattai Ravikumar and Devangini Patel .................... 49

1.7 Modalities, Styles and Strategies: An Interaction Framework for Human–Computer Co-Creativity
Anna Kantosalo, Prashanth Thattai Ravikumar, Kazjon Grace and Tapio Takala ............................................................... 57

1.8 Casual Creators in the Wild:A Typology of Commercial Generative-Creativity Support Tools
Elena Petrovskaya, Sebastian Deterding and Simon Colton ........... 65

1.9 Exploring Crowd Co-creation Scenarios for Sketches
Devi Parikh and C. Lawrence Zitnick .................................. 73

1.10 A Climate Change Educational Creator
Jun Chang and Margareta Ackerman .................................. 77

1.11 Arny: A Co-Creative System Design based on Emotional Feedback
Sarah Abdellahi, Mary Lou Maher and Safat Siddique .................. 81

1.12 Emojinating Co-Creativity: Integrating Self-Evaluation and Context-Adaptation
João M. Cunha, Pedro Martins, Nuno Lourenço and Penousal Machado 85
2

Design and CC
89
2.1 Exploring the flexibility of a design tool through different artificial
agents
Gerard Serra, David Miralles, Maria CasalsandMı́riam López . . . . 90
2.2 Deep Learning as heuristic approach for architectural concept generation
Daniel Bolojan and Emmanouil Vermisso . . . . . . . . . . . . . . . . 98
2.3 Automatic Similarity Detection in LEGO Ducks
Mark Ferguson, Sebastian Deterding, Andreas Lieberoth, Marc Malmdorf Andersen, Sam Devlin, Daniel Kudenko and James Alfred Walker 106
2.4 Evolutionary Experiments in Typesetting of Letterpress-Inspired Posters
Sérgio Rebelo and João Bicker and Penousal Machado . . . . . . . . 110

3

Language and Narrative
3.1 Rosetta Code: Improv in Any Language
Piotr Mirowski, Kory Mathewson, Boyd Branch, Thomas Winters,
Ben Verhoeven and Jenny Elfving . . . . . . . . . . . . . . . . . . . .
3.2 Creating Six-word Stories via Inferred Linguistic and Semantic Formats
Brad Spendlove and Dan Ventura . . . . . . . . . . . . . . . . . . . .
3.3 Toward Automated Quest Generation in Text-Adventure Games
Prithviraj Ammanabrolu, William Broniec, Alex Mueller, Jeremy Paul
and Mark O. Riedl . . . . . . . . . . . . . . . . . . . . . . . . . . . .
3.4 Bisociative Literature-Based Discovery: Lessons Learned and New
Prospects
Nada Lavrač, Matej Martinc, Senja Pollak and Bojan Cestnik . . . .
3.5 Paranoid Transformer: Reading Narrative of Madness as Computational Approach to Creativity
Yana Agafonova, Alexey Tikhonov and Ivan P. Yamshchikov . . . . .
3.6 Comparing Different Methods for Assigning Portuguese Proverbs to
News Headlines
Rui Mendes an Hugo Gonçalo Oliveira . . . . . . . . . . . . . . . . .
3.7 A Pragmatics-based Model for Narrative Dialogue Generation
Andreea-Oana Petac, Anne-Gwenn Bosser, Fred Charles, Pierre De
Loor and Marc Cavazza . . . . . . . . . . . . . . . . . . . . . . . . .
3.8 Creative Language Generation in a Society of Engagement and Reflection
George A. Wright and Matthew Purver . . . . . . . . . . . . . . . . .

114
115
123
131
139
146
153
161
169


3.9 Meta-level Evaluation and Transformational Creativity: An analysis of MEXICA
Juan Alvarado and Geraint A. Wiggins ................................. 173
3.10 Discovering Textual Structures: Generative Grammar Induction using Template Trees
Thomas Winters and Luc De Raedt ................................. 177
3.11 Computational Humor: Automated Pun Generation
Bradley Tyler, Katherine Wilsdon, Paul Bodily ......................... 181
3.12 TECo: Exploring Word Embeddings for Text Adaptation to a given Context
Rui Mendes and Hugo Gonçalo Oliveira ................................. 185

4 Music and Poetry ...................................................... 189
4.1 Drum Beats and Where To Find Them: Sampling Drum Patterns from a Latent Space
Alexey Tikhonov and Ivan P. Yamshchikov ............................. 190
4.2 Score and Lyrics-Free Singing Voice Generation
Jen-Yu Liu, Yu-Hua Chen, Yin-Cheng Yeh and Yi-Hsuan Yang ..... 196
4.3 Automatic Dialect Adaptation in Finnish and its Effect on Perceived Creativity
Mika Hämäläinen, Niko Partanen, Khalid Alnajjar, Jack Rueter and Thierry Poibeau .......................................................... 204
4.4 Shimon the Rapper: A Real-Time System for Human-Robot Interactive Rap Battles
Richard Savery, Lisa Zahray and Gil Weinberg ......................... 212
4.5 Being Creative: A Cross-Domain Mapping Network
Jichen Wu, Maarten H. Lamers and Wojtek J. Kowalczyk .............. 220
4.6 WeirdAnalogyMatic: Experimenting with Analogy for Lyrics Transformation
Hugo Gonçalo Oliveira ..................................................... 228
4.7 Towards balanced tunes: A review of symbolic music representations and their hierarchical modeling
Nádia Carvalho and Gilberto Bernardes .................................. 236
4.8 EMILY: An Emily Dickinson Machine
Juliana Shihadeh and Margareta Ackerman .............................. 243
4.9 Automated Music Generation for Visual Art through Emotion
Xiaodong Tan and Mathis Antony ........................................... 247
4.10 Melody Similarity and Tempo Diversity as Evolutionary Factors for Music Variations by Genetic Algorithms
Fernando H. Calderon Alvarado, Wan-Hsuan Lee, Yen-Hao Huang and Yi-Shin Chen .................................................. 251

4.11 ERwEM: Events Represented with Emotive Music Using Topic-Filtered Tweets
Makayla Harris, Hunter Harris and Paul Bodily .......................... 255

4.12 Creating Latent Spaces for Modern Music Genre Rhythms Using Minimal Training Data
Gabriel Vigliensoni, Louis McCallum and Rebecca Fiebrink ...... 259

4.13 Emotive Music Composition from Visual Sources of Inspiration
Dylan Lasher, Tyler Hedgepeth, Nickolas Nathan Taylor and Paul Bodily263

5 Performance

5.1 Show, Don’t (Just) Tell: Embodiment and Spatial Metaphor in Computational Story-Telling
Philipp Wicke and Tony Veale .............................................. 268

5.2 Creativity Metrics for a Lead-and-Follow Dynamic in an Improvisational Dance Agent
Meha Kumar, Duri Long and Brian Magerko ............................ 276

5.3 Towards Movement Generation with Audio Features
Benedikte Wallace, Charles P. Martin, Jim Torresen and Kristian Nymoen ................................................................. 284

5.4 Creativity Theatre for Demonstrable Computational Creativity
Simon Colton, Jon McCormack, Michael Cook and Sebastian Berns . 288

5.5 Feel The Music: Automatically Generating A Dance For An Input Song
Purva Tendulkar, Abhishek Das, Aniruddha Kembhavi and Devi Parikh292

6 Philosophy and Evaluation

6.1 A Leap of Creativity: From Systems that Generalize to Systems that Filter
Porter Glines, Brandon Biggs, and Paul M. Bodily .................. 297

6.2 Action Selection in the Creative Systems Framework
Simo Linkola, Christian Guckelsberger and Anna Kantosalo .... 303

6.3 Humans in the Black Box: A New Paradigm for Evaluating the Design of Creative Systems
Brad Spendlove and Dan Ventura .......................................... 311

6.4 Artificial Creative Intelligence: Breaking the Imitation Barrier
Rowland Chen, Roger B. Dannenberg, Bhiksha Raj, and Rita Singh . 319
6.5 Post-creativity and AI: Reverse-engineering our Conceptual Landscapes of Creativity
Jan Løhmann Stephensen ............................................. 326
6.6 Explainable Computational Creativity
Maria Teresa Llano, Mark d’Inverno, Matthew Yee-King, Jon McCormack, Alon Ilsar, Alison Pease and Simon Colton .......... 334
6.7 On the Machine Condition and its Creative Expression
Simon Colton, Alison Pease, Christian Guckelsberger, Jon McCormack and Teresa Llano ............................................. 342
6.8 A Study on Reproducibility in Computational Creativity Research
Martín Marzidovšek and Senja Pollak ........................................ 350
6.9 Extending the Philosophy of Computational Criticism
Jesse Roberts and Douglas H. Fisher ........................................ 358
6.10 On the Inherent Creativity of Self-Adaptive Systems
Simo Linkola, Niko Mäkitalo and Tomi Männistö ........................... 362
6.11 Towards Enhanced Creativity in Interface Design through Automated Usability Evaluation
Snehal Dhengre, Jayant Mathur, Farzaneh Oghazian, Xiaomei Tan and Christopher McComb ............................................. 366
6.12 From Computational Creativity to Creative Problem Solving Agents
Evana Gizzi, Lakshmi Nair, Jivko Sinapov and Sonia Chernova ... 370
6.13 Creativity explained by Computational Cognitive Neuroscience
Frederic Alexandre ........................................................ 374
6.14 Mexican International Colloquium on Computational Creativity
Rafael Pérez y Pérez ..................................................... 378
6.15 Printing Error: Creating Uniqueness
Hyemin Bae, Bryan Ortega-Welch and Jose Luis García del Castillo y López ................................................................. 382
6.16 Using Adaption-Innovation Theory to Simulate Robustness in Design Teams
Noriana Radwan and Christopher McComb ........................................ 386
6.17 Limits Theorems for Creativity with Intentionality
Lav R. Varshney .......................................................... 390
6.18 A Deep Dive Into Exploring the Preference Hypervolume
Alexander Hagg, Alexander Aestroth, Thomas Bäck ..................... 394
6.19 The societal and ethical relevance of computational creativity
Michele Loi, Elenora Viganó and Lonneke van der Plas ................. 398
6.20 Can a Robot Do a Trust Fall? Absurdity as a Component of Human Intelligence and Embodiment
Amy LaViers and Ilya Vidrin ............................................. 402
6.21 Bridging Generative Deep Learning and Computational Creativity
Sebastian Berns and Simon Colton ........................................ 406
6.22 Ever-changing Flags: Impact and Ethics of Modifying National Symbols
João M. Cunha, Pedro Martins and Penousal Machado ............. 410

7 Visual Creativity 414
7.1 KaoKore: A Pre-modern Japanese Art Facial Expression Dataset
Yingtao Tian, Chikahiko Suzuki, Tarin Clanuwat, Mikel Bober-Irizar,
Alex Lamb and Asanobu Kitamoto ........................................ 415
7.2 Aesthetic Preferences of Neural Style Transfer-Generated Portrait Images: An Exploratory Study with the Two-Alternative-Forced-Choice Task
Chaehan So ................................................................. 423
7.3 Exploring CC in XR: Visualizing Creative Conversation Topics to Facilitate Meaningful Face-to-Face Interaction
Hunter Harris, Makayla Thompson, Isaac Griffith and Paul Bodily . 429
7.4 Interactive Neural Style Transfer with Artists
Thomas Kerdreux, Louis Thiry and Erwan Kerdreux ................. 437
7.5 Let’s Figure This Out: A Roadmap for Visual Conceptual Blending
João M. Cunha, Pedro Martins and Penousal Machado ............. 445
7.6 Lemotif: An Affective Visual Journal Using Deep Neural Networks
X. Alice Li and Devi Parikh ................................................. 453
7.7 3D Topology Transformation with Generative Adversarial Networks
Luca Stornaiuolo, Nima Dehmamy, Albert-László Barabási and Mauro Martino ................................................................. 461
7.8 Hello, An Interactive Cinematic Generative Artwork
Pedro Alves da Veiga ......................................................... 469
7.9 Which type is your type?
Jéssica Parente, Tiago Martins, João Bicker and Penousal Machado 476
7.10 Predicting A Creator’s Preferences In, and From, Interactive Generative Art
Devi Parikh ................................................................. 484
7.11 Do Digital Agents Do Dada?
Gunter Loesel, Piotr Mirowski and Kory W. Mathewson ............ 488
7.12 Neuro-Symbolic Generative Art: A Preliminary Study
Gunjan Aggarwal and Devi Parikh ........................................ 492
7.13 Creative Constellation Generation: A System Description
Andres Sewell, Andrew Christiansen and Paul Bodily ................ 496
7.14 Pokérator - Unveil your inner Pokémon
   Dominique Geissler, Elisa Nguyen, Daphne Theodorakopoulos and Lorenzo Gatti
   ........................................................................................................ 500
7.15 An Approach for Text-to-Emoji Translation
   Philipp Wicke and João M. Cunha .................................................. 504
7.16 Hand-Crafting Neural Networks for Art-Making
   Erik Ulberg, Daniel Cardoso Llach and Daragh Byrne ..................... 508
Understanding and Strengthening the Computational Creativity Community: 
A Report From The Computational Creativity Task Force

João M. Cunha¹, Sarah Harmon², Christian Guckelsberger³, Anna Kantosalo⁴, 
Paul M. Bodily⁴ and Kazjon Grace⁵

¹Department of Informatics Engineering, CISUC, University of Coimbra, Coimbra, Portugal
²Department of Computer Science, Bowdoin College, Brunswick, USA
³Department of Computer Science, Aalto University, Espoo, Finland
⁴Department of Computer Science, Idaho State University, Pocatello, USA
⁵School of Architecture, Design and Planning, The University of Sydney, Sydney, Australia

jmacunha@dei.uc.pt, taskforce@computationalcreativity.net

Abstract

The field of computational creativity (CC) is rightfully home to a great diversity of perspectives and disciplines. As members of such a diverse community we need to pay special attention to how we can sustain a common identity, and how we communicate, support, and evaluate our work. Here, we introduce the first report from the Computational Creativity Task Force, which was established to support the collective advancement of the computational creativity research community. As a first step towards that goal, we present an exploration of who we are as a community: the authors and program committee members of the International Conference on Computational Creativity. We welcome a discussion of both this data and the mission of the task force going forwards.

Introduction

Since the emergence of computational creativity (CC) as a research field, its meaning and goals have been subject to an ongoing debate. CC approaches have been applied not only in new application areas (Loughran and O’Neill, 2017) but to entirely new kinds of creativity. Far from the niche interest we may have started as, the exploration of creativity and AI is exploding around us. Complementary research is being proposed and conducted in many other communities, e.g. on “constructive machine learning” and “machine learning for creativity and design” at NeurIPS, on “diversity and serendipity” at RecSys and SIGIR, on “imagination machines” at AAAI (Mahadevan, 2018), and on “open-endedness” in ALIFE (Stanley, Lehman, and Soros, 2017). We think that the almost twenty year history of the International Conference on Computational Creativity (ICCC) and the joint workshops that preceded it (Boden, 2015) have a lot to offer these emerging disciplines. We can see ourselves and our interests in each of those areas, but do they see us?

We are, now more than ever, champions of an idea whose time has finally come. The Computational Creativity Task Force was formed by the Association for Computational Creativity (ACC)¹ with a simple question in mind: what are we going to do about it? Our opinion is that to participate in the growing debate will bring interest and impact to our work, while to ignore it and be comfortable in our relative isolation would be a truly unfortunate missed opportunity. We propose not that we change who we are, but how we communicate: how do we speak to these parallel disciplines, not as a way to join them but as a way to give our ideas the voice needed to influence them? How can we act as ambassadors, encouraging researchers from those adjacent communities to read our work and see its value to their own? And what can we in turn learn from these communities and embrace in our work (Cook and Colton, 2018)?

This means a re-evaluation of what we conceive as computational creativity research, and how inclusive we want it to be with respect to other areas of AI and beyond. It also means a re-evaluation of how we disseminate our work, as the responsibility of building bridges to other communities cannot solely fall on those who already have one foot on both shores. At the same time, we cannot lose sight of what makes us unique: while encouraging cross-pollination and integrating with neighbouring communities, we need to retain a focus on the questions and explorations that are not being performed elsewhere. As Don Barnes would say, we’ve got to hold on loosely, but not let go.

The Computational Creativity Task Force was founded to support the field of CC and its researchers by providing a robust pipeline for strengthening community unity and productivity. It has been initiated by Christian Guckelsberger, João Miguel Cunha, Carlos Léon, and Pablo Gervás, inspired by conversations with Tony Veale, Amílcar Cardoso, Hannu Toivonen, Rafael Pérez y Pérez and others. Those discussions, which took place at the 2019 Dagstuhl workshop “Computational Creativity Meets Digital Literary Studies” (Besold et al., 2019) concerned the state of the CC community and the conference: was it still relevant when similar-sounding research attracted much larger crowds elsewhere? After so long trying to get these ideas to go mainstream, were we in danger of the mainstream passing us by? Are we as researchers sufficiently equipped to push this field forward? These questions have been raised...
by the community before, but with few practical outcomes. The task force was officially formed following a presentation of the above concerns to the ACC Steering Committee at ICC’19, and announced at the community meeting that followed that event. In an effort to be most representative of CC researchers, the present members were recruited not just based on their capacity to contribute, but also their diversity in terms of gender, experience, and research focus. The authors of this paper are the task force’s first members.

The task force operates independently of, but in close interaction with, the ACC steering committee: its members identify issues concerning the CC community, the ICC conference, the journal, and the other activities of the association. They then develop proposals to overcome these issues that are put to the Steering Committee as a whole. If accepted, these proposals are further refined and implemented.

We are dedicated to keeping the task force open, and welcome everyone to contribute their ideas and their time in any way that they can. A major goal of this report is to make our work more transparent, and encourage the community to participate in this endeavour.

This report summarises our tasks and their results so far. It builds on a body of work that is deeply concerned with the CC community, studying which lessons for future growth and knowledge exchange can be learned from neighbouring communities (Cook and Colton, 2018), investigating the diversity of application areas within CC research (Loughran and O’Neill, 2017), and examining how the field is perceived (Harmon and McDonough, 2019).

**Goals and Tasks**

The Computational Creativity Task Force was founded to foster the development of the CC field and community. However, these are abstract goals; thus, upon its foundation we identified concrete areas to address. The following goals were defined: (i) to reflect on the attractiveness of ICC as a venue for CC research and implement measures to foster openness and preserve interdisciplinarity; (ii) to assess the perception (our own, and that of others) of CC as a community and identify possible actions to improve our image; (iii) to analyse our current way of functioning, specifically regarding scientific practices, and propose changes and procedures to ensure consistent scientific quality; and (iv) to actively seek opportunities to connect with other researchers and stimulate collaboration.

With these goals in mind, we have been working on five core tasks over the past year. In the following sections, we provide a brief overview of each task.

**Community Survey**

To gain a better understanding of the needs and perceptions of the CC community, we developed a survey to distribute among regular and likely new conference attendees. Our objectives were to identify those who were interested in the community, to better understand their backgrounds and needs, and to hear their perceptions of CC as a field, research venue, and research community. As of this moment, more than 140 participants have completed the online questionnaire, recruited via email (using the CC mailing list\(^2\)), the CC forum\(^3\), Twitter, and other social media channels. We will report on the results of this survey in future work.

**Calls for Papers and Tutorials**

As a conference, ICC has sought to be seen as a driving force for CC and be a central venue to discuss the developments, goals, and future of CC-related research. While much attention has been given to CC and new perspectives have appeared, our positioning has remained mostly unchanged—an example is the Call for Papers (CfP), which has had little changes in the past years.

After reviewing the existing CfP, we identified two aspects to work on: (i) the inclusion of new areas while still providing a clear indication of the scope and (ii) an adjustment to the presentation of the CfP to attract fresh perspectives and to increase a sense of openness.

We put forward suggestions for revised ICC calls for papers and tutorials that were eventually incorporated by the program chairs for the 2020 conference. Most significantly, we proposed *Computational Creativity Translations*, a new extended short paper category. This category acknowledges that research relevant to the core CC goals is presented and published at other venues without being assessed with respect to these goals. Further, it allows for such work to be re-evaluated with respect to CC research. In this way, the category has the potential to welcome and invite more members into the community, and to make important findings transparent that would otherwise have remained unnoticed. Encouragingly, a substantial portion of tentative 2020 conference attendees have indicated they were more inclined to submit to ICC as a result of this change—this is a preliminary result obtained with the community survey. Eventually, five submissions to ICC’20 were made in this category, mostly by authors who have not previously submitted to the conference.

**Best Practices for Reviewing Papers**

As previously mentioned, one of our goals is related to maintaining the scientific quality of the research initiatives organised by the ACC. Paper reviews contribute directly to improving the quality of CC research and reinforcing the fabric of our community. Moreover, they reward, encourage, and improve the efforts of individual researchers submitting their work, and can make them feel welcome within a research field to which they may not have previously contributed. To increase these beneficial effects, we first surveyed reviewing guidelines from related conferences and developed a set of best practices for reviewing ICC papers. These guidelines were provided to the ICC’20 reviewers and were also posted on the ICC website\(^4\).

---

\(^2\)https://mailchi.mp/b3bd32bb89e3/cccmailinglist

\(^3\)https://groups.google.com/forum/#!forum/computational-creativity-forum

\(^4\)http://computationalcreativity.net/iccc20/reviewer-best-practices/
Code of Conduct

Professional conferences and organisations typically have a code of conduct that specifies acceptable versus unacceptable behaviour in relation to member activities, procedures for reporting harassment, and addressing grievances (e.g., Ruby Berlin e.V., 2020). In an effort to prevent such unacceptable behaviour and to highlight ICCC as a safe and inclusive space to regulars and newcomers, we have developed an initial draft of a formalised ICCC Code of Conduct to be implemented in future years. Crucially, we do not consider such guidelines a way to make parliahs out of the un-reformed, but should rather remind us of the fact that we all have biases and prejudices. It is more important that we develop good guidelines rather than coming up with them fast, and work on this task is thus ongoing.

Diversity Strategy

We are dedicated to increasing the diversity and inclusiveness of the CC community and ICCC as its main conference. More specifically, we want to welcome researchers representing a wide variety of, amongst others, cultures, races, ages, and gender identities, with a particular focus on traditionally disadvantaged groups. Moreover, we are concerned with strengthening diversity in expertise and functional background, e.g., in terms of the research field that an individual identifies with most in order to support CC as an interdisciplinary research endeavour. In order to devise effective strategies to be more inclusive and increase as well as sustain diversity, we analysed the diversity of the CC community. Our work on this task is ongoing, and we describe preliminary results on the diversity in (i) conference authors and submissions and (ii) program committee members in this report.

First Steps Toward Better Understanding the ICCC Community

To inform the previous tasks and to better understand the active ICCC community more generally, we gathered statistics relating to published papers, their authors, and the program committees (PCs) of different years. Here we focus primarily on 2020, as the full statistics are still being gathered.

The main source of our statistics are the proceedings of the conference which include, in addition to the papers and their authors, a list of PC members and their affiliations. The ICCC proceedings provide the foundation for the data which we used, augmented, and evaluated in this paper. In compliance with the European Union’s General Data Protection Regulation, we removed the names of authors and PC members in our datasets. We moreover categorised the data and calculated aggregate statistics to improve anonymity.

Results

Analysis of Conference Authors and Submissions

We conducted an analysis of the authors submitting to ICCC since its inception in 2010. The total number of different authors was 706. In order to analyse this data, we established three time periods: the first five years of the conference series (2010-2014) and two periods from the last 6 years (2015-2017 and 2018-2020). These periods were used to divide the authors into groups according to their publication behaviour—i.e., whether they published papers in these periods or not (see Figure 1). From the results we can see two significant groups: G1—authors that only published in the first five years of the conference—and G7—authors that only published in the last three. The value of this latter group (245) is an indication that the conference is attracting new people. G5 is also worth mentioning as it consists of authors that only published in the period 2015-2017 and were not retained for the following years. In contrast, G6 are authors who first published in the conference during the 6th-8th years and continued publishing (have accepted papers in the last three years). Additionally, there are 44 authors who have published in the first five and in the last three (G3+G4) years of the conference.

In order to further study the growth of the conference in recent years, we have focused on new authors and their retention (see Table 1). The data collected concerns the number of new authors in the papers (short and full) that were submitted and accepted in the last three years. Though data on accepted papers is available for all three years, we were only able to obtain data regarding submissions (both accepted and rejected) for 2020. Table 1 indicates an increasing trend in the number of different authors on individual papers, and this trend also occurs in the percentage of new authors—43.8% in 2018, 58.2% in 2019 and 70.9% in 2020. Something that clearly stands out is the increase of 107.46% in the number of new authors from 2019 to 2020 (67 vs 139, respectively). A similar increase is visible in comparing the number of full papers submitted in 2019 (52) with those from 2020 (80). Even though we cannot be certain of the reason behind this substantial increase, we hypothesise that it was due to a combination of an improved...

<table>
<thead>
<tr>
<th>Authors</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>G1</td>
<td>209</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G2</td>
<td>51</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G3</td>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G4</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G5</td>
<td>122</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G6</td>
<td>35</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G7</td>
<td>245</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>706</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1: ICCC 2010-2020 author groups. The figure shows the number of authors of seven groups (G1-G7) based on three time intervals (marked in blue, green and orange): the first five years of the conference (2010-2014), years 6 to 8 (2015-2017), and the last 3 years (2018-2020). For example, G1 are authors that only published in the first five years.
Call for Papers, as detailed earlier, and a more aggressive dissemination process.

By analysing the papers, it is possible to identify the context of submission by new authors. Table 2 summarises the results of our analysis in terms of new versus returning authors in the years of 2019 (full and short papers) and 2020 (full, short, and workshop papers), for submitted papers (S) and accepted papers (A). The table shows the percentages of papers with all recurrent authors (R), i.e., papers in which all authors are returning authors, papers with at least one new and one recurring author (RN), and papers with authors that are all new to ICCC (N). The label #P refers to the total number of papers. Even though the analysis in the previous paragraphs did not concern workshop papers, we decided to include them in the analysis of the submissions. Overall, we find that submissions with all recurrent authors comprised only 25% of the 2019 full paper submissions and 22.5% in 2020—which increased to 32.56% of accepted papers. There were 44.23% and 46.25% full paper submissions by all new authors for 2019 and 2020, respectively. We observe the lowest percentages of all recurrent authors for short papers, and the highest percentages of all new authors in workshops. Another interesting fact is the high percentages of mixed recurrent and new authors (RN); they highlight that contributors to past editions tend to bring new people to the conference rather than working only with other recurring authors. Table 1 also shows the numbers of retained authors in 2019, 2020 and a combination of 2019-20, which translate into retention rates of 17.9%, 14.9%, and 18.8%, respectively.

### Analysis of the Program Committee

As a starting point for analysing trends in the composition of community leadership, we analysed and compared the diversity of regular and senior PC members in the 2020 instance of ICCC.

### Data Collection

We have considered diversity in terms of gender, place of work, professional experience, and affiliation with academia or industry. We considered the members’ job titles as proxy for the latter two categories. We gathered the corresponding data using the following methods:

- **Gender**: For our purposes, the term ‘gender’ does not directly refer to either the sex of the author (at birth or chosen later) or the gender of the author (socially assigned or self-chosen). Instead, ‘gender’ in our analysis was determined based on the gendered word usage associated with an individual on public websites (such as their institution-specific profiles or biographies featured on a research paper). For example, a PC member with an online biography containing the pronouns “he” and/or “his” was assigned a gender of male. Since even institutional pages may miss gender, these data do not necessarily reflect people’s chosen identity, and are thus only an estimate.

- ** Continent of Work**: The geographic location of each PC and senior PC member was assigned based on their reported work institution. This data does not speak to the full cultural background of the people involved in the committees, but reflects in general on the geographic balance of the committee. In order to arrive at a set of useful, but general categories, we used continents as markers for different categories based on the seven continent model. Where members have affiliations with institutions across multiple continents, we considered each affiliation separately.

- **Experience and Affiliation with academia or industry**: The professional experience of the committee members was inferred from their current job title, gathered from publicly available sources. We used Google Scholar and LinkedIn profiles, and finally institutional pages when the former two were unavailable or did not list the required information. No information could be found for three PC members, and they were thus omitted from our evaluation. Academic roles were categorised as per the defacto standard used in universities from the United States. On average, we expect researchers from an arbitrary country most likely to be familiar with this than with any other rank system. Moreover, the US system has arguably a very

### Table 1: Analysis of new authors. The number of new authors and total authors both in papers submitted and accepted (including full and short papers). The table also shows the number of authors retained, i.e., new authors that published in a subsequent year. We only considered the three years from 2018 to 2020 and the two-year interval from 2019-20. The 2019-2020 number indicates the sum of new authors in 2018 with submissions in 2019 or 2020 and new authors in 2019 with submissions in 2020. Values that we were not able to collect are marked with .

<table>
<thead>
<tr>
<th></th>
<th>submitted</th>
<th>accepted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>new total</td>
<td>new total</td>
</tr>
<tr>
<td>2018</td>
<td>39 89</td>
<td>7</td>
</tr>
<tr>
<td>2019</td>
<td>67 115</td>
<td>10</td>
</tr>
<tr>
<td>2020</td>
<td>230 299</td>
<td>20</td>
</tr>
<tr>
<td>2019-20</td>
<td>139 196</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 1: Analysis of new authors. The number of new authors and total authors both in papers submitted and accepted (including full and short papers). The table also shows the number of authors retained, i.e., new authors that published in a subsequent year. We only considered the three years from 2018 to 2020 and the two-year interval from 2019-20. The 2019-2020 number indicates the sum of new authors in 2018 with submissions in 2019 or 2020 and new authors in 2019 with submissions in 2020. Values that we were not able to collect are marked with .
low resolution, and its use thus decreases the chances for misclassification. We furthermore used these job titles to distinguish roles in academia and industry.

Findings: Gender Distribution

We found a strong gender inequality in the two program committees, with members classified as female being in an even larger minority in the senior PC. Across both committees, less than one third of all members have been classified as female. Figure 2 illustrates the corresponding ratios.

Findings: Continent of Work

We found that some continents, despite being home to institutions that make strong contributions to research, are severely underrepresented. This is most striking for Asia, which is only represented by three regular PC members. Australia and South America both have four regular PC members, but Australia is also represented by two senior members. Crucially, Asia and South America are not represented in either committee. The two continents with the most representation in both committees are North America and Europe. Over 50% of members in both committees work at European institutions. A breakdown of PC members by continent of work is presented in Figure 3.

Findings: Professional Experience

The regular and senior committees are both primarily comprised of researchers at the assistant professor level or above. Regular PC members are typically only recruited into the senior PC after having collected substantial research experience beyond their PhD. This is reflected in the absence of PhD students and postdocs in the senior PC. Our findings, as illustrated in Figure 4, do not reveal any unexpected diversity issues given this recruitment policy. We do, however, find that the vast majority of members in both committees have academic roles, although a substantial amount of relevant work which could be submitted to ICCC is nowadays conducted in industry. The latter ratios are highlighted in Figure 5.

Findings: Paper Submissions

We also crossed the data from the PC with data from the conference papers (previously described) to assess how the members of the PC are contributing to the conference. We analysed four groups of people: (i) members of the ICCC’20 PC, (ii) members who have been part of the PC since 2019 and (iii) since 2020, and (iv) members who were once in the PC but have not been a member in the past 3 years (no longer PC). For each of these groups, we present the number of people that have published at least one paper in 4 time periods (ICCC’20, past 3 years, past 5 years and never). The results can be found in Table 3. We notice that only 25 members of the current PC have at least one accepted paper in ICCC’20 and 47 in the last 3 years. Even more striking is that 15 PC members have never published in ICCC. We note though that this includes 9 people who have only joined the PC in the last 2 years. This number may be alarming if one considers that a PC member is supposed to contribute with their own work. On the other hand, it indicates that there is space for people who may be good assets to the PC even though they do not publish. Another interesting result is that people recently added to the PC have already contributed with papers to the conference. Moreover, some of the people who are no longer members of the PC continue to publish.

Discussion and Future Work

Our work so far has shown that there are improvements to be made regarding two different aspects: scientific and social. Regarding the latter, based on the initial results of our PC analysis, we highlight the need to establish and sustain gender identity equality across both the regular and senior PC. Based on the results reported by Wang et al. (2019), the average proportion of female authors in computer science is currently 27%. Even though this is not directly comparable
to the PC composition, it suggests that some effort has been made to achieve the 32% (see Figure 2) assumed gender ratio in the ICCC’20 program committee (regular and senior). Nonetheless, more work needs to be done to identify and include emerging non-male-identifying researchers. Moreover, we recommend recruitment of more members from presently underrepresented continents, in particular from Africa, Asia, and South America. To this end, it might be worthwhile to consider whether some of the present Asian or South American regular PC members would be suitable for a senior role.

When it comes to the scientific aspect, several topics should be addressed. First, we should improve the perception of ICCC as a welcoming venue. We have already made progress toward this goal in the form of our proposed changes to the Call for Papers (specifically the CC Translations category) which have successfully attracted new authors. Similarly, we should keep promoting and make good use of the short paper and workshop tracks as they have been shown to come with the highest percentages of new authors. Our analysis of 2019 and 2020 ICCC submissions revealed an initial picture of returning versus new authors. Nonetheless, it will be important to assess the causes for authors to move away from ICCC. More analysis will be needed in future years, especially since the 2020 pandemic might have extraordinarily affected conference submissions.

It is not only important to attract new people but also to give them the opportunity to actively contribute to the community early on. One way of facilitating this is to increase the number of PhD students and postdocs in the regular PC. Additionally, since more and more relevant research contributions are submitted to ICCC from industry, it would be desirable to have more industry representatives in both committees to ensure that these contributions receive construc-

Figure 4: PC members’ professional experience, based on job titles mapped to the US Rank System. Academic and industry postdocs and researchers are jointly considered. We only took into account primary jobs, and left out three PC members for whom job information could not be obtained.

Table 3: Paper contributions by PC members. The table shows the number of people in regards to PC membership and ICCC paper publication. The categories presented are: 2020 refers to people with at least one paper in 2020; 3 years refers to the number of people with at least one paper in the past 3 years (2018-2020); 5 years refers to people with at least one paper in the past 5 years (2016-2020); and never refers to people who never had a paper in ICCC. We present results for four groups of people: members of ICCC’20 PC; people who have joined the PC in 2019 (since 2019) and in 2020 (since 2020); and people who were at least once in the PC but have not been a member in the past 3 years (no longer PC). T is the total of people in each group.

Figure 5: Academic and industry representatives in PCs. Three members without job information were left out.
tive feedback and appreciation. As we identified in our analysis, the existence of PC members who have not yet published is an indication that there is space for people who may be good assets to the PC even though they do not publish often. We should carefully consider though what these members bring to the table, e.g., in terms of valuable expertise from adjacent areas. One example is artists, who could certainly bring a valuable perspective to the review process. We should also identify former PC members that are still actively contributing to the conference, and carefully consider the merits of inviting these individuals to rejoin the PC.

The first step towards including both young, ambitious researchers and representatives of other fields and professional domains (e.g., industry) is to design a more systematic procedure for recruiting members to the ICCC PCs. This policy should not only value scientific contributions but also an individual’s potential to stimulate ICCC’s interdisciplinarity.

In general, we believe that there are great opportunities for growth, and ICCC should continue to seek its place as a central venue of the CC field. To this end, we should not only seek to maintain and increase the quality of research, but also foster interdisciplinarity and diversity through the implementation of concrete measures to reach under-represented groups and communities.

Acknowledgments

AK is funded by the Academy of Finland grant 311090 (Digital Aura). CG is funded by the Academy of Finland Flagship program “Finnish Center for Artificial Intelligence” (FCAI). JMC funded by the FCT - Foundation for Science and Technology, I.P., within the scope of the project CISUC - UID/CEC/00326/2020 and by European Social Fund, through the Regional Operational Program Centro 2020, and under the grant SFRH/BD/120905/2016.

References


1. Co-Creativity and Support
Shaping the Narrative Arc: Information-Theoretic Collaborative Dialogue
Kory Wallace Mathewson1*, Pablo Samuel Castro2, Colin Cherry3,
George Foster3, Marc G. Bellemare2

Abstract
We consider the challenge of designing an artificial agent capable of interacting with humans in collaborative dialogue to produce creative, engaging narratives. Collaborative dialogue is distinct from chit-chat in that it is knowledge building, each utterance provides just enough information to add specificity and reduce ambiguity without limiting the conversation. We use concepts from information theory to define a narrative arc function which models dialogue progression. We demonstrate that this function can be used to modulate a generative conversation model and make it produce more interesting dialogues, compared to baseline outputs. We focus on two modes of modulation: reveal and conceal. Empirically, we show how the narrative arc function can model existing dialogues and shape conversation models towards either mode. We conclude with quantitative evidence suggesting that these modulated models provide interesting and engaging dialogue partners for improvisational theatre performers.

Introduction
Designing and building computational models that generate meaningful dialogue for human-interaction is a challenging open problem. Conversational agents can be effective for health-care (Bickmore and Giorgino 2006), by supporting cognitive-behavioural therapy for treating depression (Fitzpatrick, Darcy, and Vierhile 2017), and supporting reminiscence (Nikitina, Callaioli, and Baez 2018) if they are capable of interaction and collaboration.

Rule-based conversational models have existed for over 50 years (Weizenbaum 1966). These methods are limited by hand-tuning and engineering to predict and handle possible inputs. Conversely, generative language models maximize the likelihood of an utterance (e.g. a sentence or sequence of words) (Graves 2013). These models can predict the likelihood of an utterance by considering sentences as a sequences of words or tokens. This objective generates grammatically correct and semantically related to surrounding context it, but lack global consistency (Liu and others 2018).

What makes some dialogues more interesting than others? Interesting collaborative dialogue constructs knowledge iteratively (Swain 2000) and depends on each speaker bringing information to the conversation (Sawyer 2003). Interestingness is subjective and difficult to directly optimize via numerical methods (Li and others 2016a).

Our work uses a narrative arc to incrementally construct shared knowledge. A narrative arc defines evolving qualities of emotion, tension, or topic over a story. We draw inspiration from improvised theatre, where actors collaborate in real time to develop narrative based on thematic constraints (Johnstone 1979). Improvised theatre is a unique storytelling medium which relies on collaborative dialogue in which each utterance contributes significant information (Swain 2000). We appeal to the two golden rules of improvised dialogue, characteristic of interesting collaborative dialogue (Johnstone 1979; Sawyer 2003). Good dialogue should 1) accept (i.e. be consistent with the dialogue thus far and 2) reveal (i.e. progress the dialogue with new information).

In this work, we propose a new method to modulate a conversation model, which accepts input utterances by generating consistent and revealing responses. Our approach combines a conversational model with a topic classifier, which we call a universe model. We borrow the term universe from improvised theatre where it is used to describe the world-as-we-know-it (Johnstone 1979; Raby 2010). A universe encompasses associations in the dramatic world and is motivated by the possible world semantics theory (Kripke 1963).

We identify two modes of operation for our shaping method: revealing and concealing. Revealing dialogue adds additional information about the current universe. Generating utterances which progress a scene with new information is the primary goal of our approach. Concealing dialogue avoids exposing new information about the universe. The ability to generate both revealing and concealing dialogue is a convenient side-effect of this method.

The universe model characterizes the information revealed by each utterance in a sequence. We refer to this information profile across utterances as the narrative arc. By tuning how revealing the model is, we selectively choose utterances to shape the narrative arc to produce more interesting and engaging dialogue. We argue that a balance between revealing and concealing is required for interesting and engaging collaborative dialogue; both over-specification and ambiguity are undesirable. We hypothesize that there is an ideal region of information revelation which our method can expose in existing text-based narratives such as movie scripts.
Shaping the Narrative Arc

In this section, we present a mechanism for shaping the narrative arc inspired by combining methods exploring entropy in textual documents (Shannon 1951) with the *Simple Shapes of Stories* described by Vonnegut.\(^1\) We describe concepts of conversation and universe models. Then, we show how these combine to describe a narrative arc. Finally, we show how the narrative arc can be used to generate interesting dialogue.

The Conversation Model

A conversation model accepts an input utterance and generates one, or several, output utterance(s). The model maintains local coherence by conditioning output generation on the input. We write \(\mathcal{X}\) to denote the set of possible utterances (i.e. sequences of words); in this work, \(\mathcal{X}\) is a collection of English sentences. A sequence of \(t\) successive utterances is a dialogue, denoted \(x_{1:t}\). A conversation model yields probability \(q\) of utterance \(x_t\) given dialogue \(x_{1:t-1}\).

We focus on dialogue generation using three retrieval-based conversation models. The first two models are based on the OpenSubtitles dataset (Lison, Tiedemann, and Kouylekov 2018). When queried with an input line \(x_{t-1}\), a model returns \(K\) candidate responses:

- **Baseline Random model**: sample \(K\) lines from \(\mathcal{X}\).
- **Deep neural network model (DNN)**: we embed all the lines in \(\mathcal{X}\) into a latent semantic space \(\mathcal{S}\) using the Universal Sentence Encoder (Cer and others 2018). We encode the input line into \(\mathcal{S}\), and return the \(K\) approximate nearest neighbours in \(\mathcal{S}\) using the \(L^2\) norm as the distance metric. Similar to the DNN model, a third model (Books), responds with semantically related nearest neighbour lines from literature, filtered for offensive content.\(^2\)

The Universe Model

The universe model measures how each successive utterance of a dialogue influences the probability distribution over universes. For a given utterance, the universe model calculates a probability distribution over universes. For a sequence of utterances, we use recursive universe belief propagation to update the posterior over the course of a dialogue. Revealing dialogue would concentrate probability mass on a single universe, and concealing dialogue would maintain posterior likelihood over a set of universes. The shape of this sequence of posteriors is the narrative arc. We investigate reveal and conceal dynamics using three different universe models based on probabilistic topic classifiers.

- **Newsgroups**: Using the newsgroup classification dataset, we filter out stop-words, create frequency vectors, and use the TF-IDF (term frequency / inverse document frequency) word weighting scheme to account for word importance in the corpus. We train a naïve Bayes classifier on 5 aggregate topic universes (**COMPUTERS**, **RECREATION**, **RELIGION**, **SCIENCE**, and **TALK**).
- **Movies**: naïve Bayes classifier, trained similar to Newsgroups, using a collected dataset of film synopses and one of 10 corresponding genres (**Drama**, **Comedy**, **Horror**, **Action**, **Crime**, **Romantic Comedy**, **Romance**, **Thriller**, **Film Adaptation** and **Silent Film**) from Wikipedia data (Hoang 2018).

- **DeepMoji**: Deep neural network that takes input text and outputs a distribution over a set of 8 aggregated emoji universes: (**SAD**, **MAD**, **Meh**, **Nerous**, **Glad**, **Music**, **Love**, and **Miscellaneous**) (Felbo and others 2017). Input text is not transformed, and a pretrained model is used.\(^3\)

Recursive Universe Belief Propagation

We desire a means by which we can update the universe belief incrementally as evidence is accumulated with each successive utterance in a dialogue. We begin by defining the notion of a **universe model** as a means of modelling the dynamics of information revelation. Consider a finite set of universes, \(\mathcal{U}\). The role of a universe model is to assess the compatibility of an utterance with a given universe, \(u \in \mathcal{U}\). Given such a model, we develop a method to update the agent’s posterior universe distribution over a sequence of utterances. For each universe \(u\), the universe model assigns a likelihood \(p(x_t | x_{1:t-1}, u)\) to an utterance \(x_t\), conditioned on a dialogue \(x_{1:t-1}\).

The universe model iteratively updates a posterior belief over universes in a similar spirit to prediction with expert forecaster (Cesa-Bianchi and Lugosi 2006). The probability of a given universe depends on iteratively combining evidence in support of that universe. The posterior probability over universes \(\mathcal{U}\) given a sequence of \(t\) utterances \(x_{1:t}\) is recursively defined as:

\[
p_t(u | x_{1:t}) = p_{t-1}(u | x_{1:t-1}) \times \frac{p(x_t | x_{1:t-1}, u)}{p(x_t | x_{1:t-1})}
\]

Where \(p_{t-1}(u | x_{1:t-1})\) is prior probability, \(p(x_t | x_{1:t-1}, u)\) is likelihood of utterance conditioned on the past dialogue and universe, and \(p(x_t | x_{1:t-1})\) is likelihood of utterance under the conversation model.

Let \(p_0(u|\cdot) = 1/|\mathcal{U}|, u \in \mathcal{U}\) be an initially uniform distribution over universes (i.e. universe model’s prior). We can marginalize out the universe if the evidence is consistent over all hypotheses. To illustrate the relationship between utterance likelihood and universe, we can explicitly write the marginal likelihood as:

\[
p(x_t | x_{1:t-1}) = \sum_{u'} p_{t-1}(u' | x_{1:t-1})p(x_t | x_{1:t-1}, u')
\]

Thus, the posterior is updated recursively as:

\[
p_t(u | x_{1:t}) = p_{t-1}(u | x_{1:t-1}) \times \frac{p(x_t | x_{1:t-1}, u)}{\sum_{u'} p_{t-1}(u' | x_{1:t-1})p(x_t | x_{1:t-1}, u')}
\]  \hspace{1cm} (1)

In practice, it may be convenient to use the output \(z(u|x_t)\) of a probabilistic classifier in lieu of a likelihood function conditioned on past utterances \(x_{1:t}\) and universe \(u\). Universe classifiers can be trained separately from language models.

\(^1\)From K. Vonnegut lecture: https://goo.gl/JuEDVR
\(^2\)https://books.google.com/talktobooks/
\(^3\)github.com/bfelbo/DeepMoji
and provide complementary signal if model input distributions overlap. This assumption is justified when both models work with similar training corpus vocabularies. We view the probability distribution over universes output by the universe model as derived from a joint distribution \( z(u, x_t) \), of the universe \( u \), and utterance \( x_t \). With \( z(u) \) as the prior distribution over universes, the conditional probability is:

\[
    z(u | x_t) = \frac{z(u, x_t)}{z(x_t)} = z(u) \times \frac{z(x_t | u)}{z(x_t)}
\]

We can substitute \( z(-|x_t) \) for \( p(x_t | x_{1:t-1}, \cdot) \) in Eq. 1 by assuming conditional independence (i.e., \( p(x_t | x_{1:t-1}, u) = p(x_t | u) \)) in uniform prior distribution (i.e., \( z(u) = 1/|U|, u \in U \)) and constant marginal probability (i.e., \( z(x_t) = \sum_{u'} p(u') p(x_t | u') \)). These assumptions are justified when the probabilistic topic classifier is a naïve Bayes classifier with uniform prior (Bishop 2006). Thus, the substitution follows the following steps:

\[
    p(x_t | x_{1:t-1}, u) \approx z(x_t | u) \quad \text{[cond. independence]}
\]
\[
    = \frac{z(u | x_t) z(x_t)}{z(u)} \quad \text{[Bayes’ theorem]}
\]
\[
    \approx z(u | x_t) \quad \text{[z(u) uniform prior]}
\]
\[
    \approx z(u) \times \frac{z(x_t | u)}{z(x_t)} \quad \text{[z(x_t) const. marginal]}
\]

Then, the entropy change \( \Delta(\cdot) \) due to a new utterance, \( x_t \), given the past dialogue, \( x_{1:t-1} \), is defined as:

\[
    \Delta(x_t; x_{1:t-1}) := H(p_t(\cdot)) - H(p_{t-1}(\cdot))
\]

The term \( \Delta(x_t; x_{1:t-1}) \) measures how much a given utterance \( x_t \) changes the entropy of the posterior, given the previous utterances \( x_{1:t-1} \). A positive value of \( \Delta(\cdot) \) is a reduction in entropy (i.e. information about the universe is revealed). Conversely, a negative value of \( \Delta(\cdot) \) is an increase in entropy (i.e. concealing). We define the score of an utterance \( x_t \), with respect to a dialogue, \( x_{1:t-1} \), as:

\[
    \sigma(x_t; x_{1:t-1}) := \exp\left\{ \alpha \Delta(x_t; x_{1:t-1}) \right\}, \quad \alpha \in \mathbb{R}
\]

The exponential function is a convenient way to ensure strict positivity and preserve the ordering of scored candidates. We use our entropy-based score function \( \sigma \) to modulate the sampling of a base conversation model, \( q \), toward \( \tilde{q} \), which depends on the change in entropy due to the new utterance.

\[
    \tilde{q}(x_t | x_{1:t-1}) \propto q(x_t | x_{1:t-1}) \times \sigma(x_t; x_{1:t-1})
\]
If $\alpha = 0$, $\sigma(\cdot) = 1$ and candidates are sampled according to $\tilde{q} = q$. If $\alpha \neq 0$, $q$ is modulated by the score $\sigma(\cdot)$. Modulation mode depends on the value of $\alpha$:

- $\alpha > 0$ (reveal): modulate $q$ towards revealing the universe. The probability of utterances likely under the universe with highest probability are increased.
- $\alpha < 0$ (conceal): modulate $q$ towards concealing the universe. The probability of utterances likely under multiple unlikely universes is increased. Utterances not supporting the likely universe are made more likely.

We use these two modulations for filtering samples from our base conversation model. We filter via one of two methods for sampling from an unnormalized distribution: greedy sampling and rejection sampling. Greedy sampling scores a set of samples from the conversation model and selects the candidate with the maximum score. Scoring a large set of candidates can be time intensive. Rejection sampling (Alg. 1) can sample from the desired unknown modulated distribution online (Murphy 2012). As the entropy function is bounded, the utterance score $\sigma$ is bounded. In practice, we set a max score and weigh all utterance scores $\sigma$ above the threshold equally. Both filtering methods have benefits. Rejection sampling provides a smoother distribution and does not require scoring a large set of candidates. Greedy sampling is less sensitive to the range of $\Delta$ from different utterances.

### Evaluation

**Narrative Arc of Existing Dialogues**

In Fig. 2, we visualize the narrative arc underlying the first 20 lines of Shakespeare’s Romeo and Juliet using three universe models: 1) Newsgroups, 2) Movies, and 3) DeepMoji.

Fig. 2 illustrates the entropy-reducing nature of good dialogue by showing us the underlying, evolving, narrative arc. Under the Newsgroups universe model, the dialogue evolves toward a TALK-centric universe. Under the Movies model, the same dialogue balances between comedy and drama before shifting towards drama. Finally, using the DeepMoji universe model, a developing ambiguity between DeepMoji universes SADNESS and LOVE is uncovered. This supports the hypothesis that existing dialogues exhibit underlying narrative arcs conditioned on universe models.

**Shaping the Narrative Arc**

In this section, we demonstrate that our method is able to modulate conversation models toward generation of revealing or concealing dialogues. Linguistic quality and semantic consistency of utterances are determined by the language underlying the conversation model. We emphasize evaluation of narrative arc shaping by focusing on the information contribution of the subsequent utterances.

We use the DNN conversation model to test how preferential selection, induced by our score function, can modulate information introduced into the conversation. In Fig. 3 we present characteristic narrative arcs and dialogues using concealing (top), neutral (middle), and revealing (bottom) modes. Each generation was primed with the first two lines from Romeo and Juliet (shown in bold in Fig. 3).

A significant difference is exposed between concealing (top) which tends toward a high entropy, uniform universe...
distribution, and revealing (bottom) where drama tends toward 1.0. Drama remains the most likely universe (and visible on all plots) as it was supported by the first two lines and subsequent utterances did not significantly shift the distribution. Fig. 3 also shows the utterances selected by the model. Concealing utterances do not add information to the dialogue, revealing utterances incorporate new information over the course of the dialogue.

We next evaluate our method’s ability to filter for concealing/revealing utterances by measuring the entropy under both an objective universe (i.e. the universe model used for scoring in generation) and a test universe not used for scoring. We use the Newsgroups universe model for objective scoring and the Movies model for testing. A random conversation model is used to generate response candidates.

We generate 20 conversations following a process similar to Algorithm 1 but using greedy sampling. Each conversation starts with a random dialogue starter line to encourage diversity and then 19 lines are sampled from the conversation model using the narrative arc function. This approximates the length of a medium-duration improvised conversation (Sawyer 2003).

Results are presented in Fig. 4. There is a significant difference between the entropy under the objective and testing universes, but each model exhibits similar dynamics over the dialogues. We conclude that concealing dialogue can conceal under multiple universes, and revealing dialogue can reveal information under multiple universe models.

The revealing/concealing dynamics of each utterance may be related to measurable lexicographical qualities such as words per sentence (WPS). We analysed the language used in 190 lines from each model and found a significant difference ($p < 0.001$) between utterances selected by the revealing model (9.26 ± 5.7 WPS) and utterances selected by the concealing model (5.05 ± 2.79 WPS).

**Predicting the Next Best Line**

We next test the system’s ability to add information to improve performance on a prediction task. Given a sequence of 5 gold-standard conversational utterances and a list of 10 next utterance candidates (i.e. the ground truth and 9 distractors), can the universe model be used to improve accuracy of predicting the ground truth?

Evaluation compares top-3 accuracy and mean reciprocal rank (MRR) over samples in a held out test set. Accuracy measures the likelihood that the system scores the ground truth within the top-3 candidates against the distractors. MRR
compares average ground truth ranking across conditions. A Transformer language model was trained on OpenSubtitles (Lison, Tiedemann, and Kouylekov 2018) to predict an output given a set of input lines (Vaswani and others 2017).

The trained Transformer model was used to assign a perplexity score for output line candidates given an input context line. For each unique subtitle file in the validation and test sets, the concatenation of the first 5 lines serves as input context and line 6 is the ground truth output to be predicted. Negative candidates are randomly selected from lines in the respective corresponding data segment (i.e. validation/test), thus may not be from the same file as the input context lines.

The perplexity under the trained conversation model serves as the unmodulated probability \( q(x_t | x_{1:t-1}) \) (Eq. 3) of selection in the prediction task. The input sequence is then passed, line-by-line, through a Newsgroups universe model and a score is assigned to each candidate relative to the change in entropy of the evolving posterior. The \( \alpha \) value is modulated over 100 evenly spaced values between \([-2, 2]\). The accuracy of predicting the ground truth in the top-3 candidates and the MRR of the ground truth are computed.

The results on the validation set are shown in Fig. 5. By selecting the correct \( \alpha \) value, the likelihood of correctly selecting utterances revealing an incremental amount of information increases significantly. Note the shape of the curve as \( \alpha \) changes. As hypothesized, there exists a region, between 0 and 1 where the ‘right’ amount of universe information is revealed. This region corresponds to the notion that each line of dialogue will reveal some, but not too much, information about the universe. As \( \alpha \) continues to increase, the accuracy decreases below the neutral baseline. The top-3 accuracy of prediction increases when the universe model boosts the probabilities of appropriately revealing dialogue. The validation set is used to select the optimal \( \alpha \), which is then used to score samples in the test set and results are presented in Table 1. Two additional models are included for comparison. T2T@l uses 1 preceding the ground truth as context. Unigram assigns a perplexity to output candidates by building a unigram language model using the 5 input lines as a corpus. A smoothing factor of \( 1 \times 10^{-5} \) is used for out-of-vocabulary words. Additionally, a random conversation baseline model is included. For each model tested, information from the universe model significantly improves the predictive accuracy on this task.

### Interactive Collaborative Dialogue

Finally, as a practical implementation case-study, we tested how this system performs in collaborative dialogue through interaction with humans. 4 expert improvisational theatre performers engaged with the system in 3 text-based conversations. Each conversation consisted of 5 utterance-response pairs for a total of 10 utterances (i.e. an average length of a short-duration improvised scene (Sawyer 2003)). Subjects are native English speakers with 5+ years professional performance experience and are familiar with shared narrative development and collaborative dialogue. Each interacted with revealing, concealing, and neutral models in a randomized order unknown to the them.

This experiment used the Books conversation model and

<table>
<thead>
<tr>
<th>CM</th>
<th>UM</th>
<th>Top3Acc</th>
<th>MRR</th>
</tr>
</thead>
<tbody>
<tr>
<td>T2T@5</td>
<td>Neutral</td>
<td>0.520</td>
<td>0.456*</td>
</tr>
<tr>
<td>T2T@5</td>
<td>Neutral</td>
<td>0.507</td>
<td>0.444</td>
</tr>
<tr>
<td>T2T@1</td>
<td>Neutral</td>
<td>0.483</td>
<td>0.428*</td>
</tr>
<tr>
<td>T2T@1</td>
<td>Neutral</td>
<td>0.469</td>
<td>0.412</td>
</tr>
<tr>
<td>Unigram</td>
<td>Neutral</td>
<td>0.366</td>
<td>0.337*</td>
</tr>
<tr>
<td>Unigram</td>
<td>Neutral</td>
<td>0.296</td>
<td>0.290</td>
</tr>
<tr>
<td>Random</td>
<td>Neutral</td>
<td>0.302</td>
<td>0.294</td>
</tr>
</tbody>
</table>

Table 1: Results for predicting the next line. CM is the conversation model, UM is the universe model, Top3Acc is the accuracy of predicting the ground-truth in the top-3 of 10 candidates, and MRR is the mean reciprocal rank of the ground truth. Unigram CM calculates the perplexity for each candidate given the input lines as training corpus. T2T@N is a Tensor2Tensor Transformer model which uses the previous N lines as an input to predict the output and NG is the Newsgroups universe. A Neutral universe model represents no modulation which is equivalent to \( \alpha = 0 \). * indicates \( p < 0.05 \) for a Students’ t-test comparing MRR to the Neutral model.

The DeepMoji universe model. Following the interactions, each performer was asked the following question: “please rank the conversations from 1 (most engaging) to 3 (least engaging)”. Engagingness was defined to align with the notions of revealing and concealing in this work. An agent is engaging for shared scene development if it brings just enough information to add specificity and reduce ambiguity but not limit the conversation.

Three of the four performers ranked the revealing model, \( \alpha > 0 \), as the most engaging. Those three performers ranked \( \alpha = 0 \) as being less engaging due to being “too random”. All subjects ranked \( \alpha < 0 \) as being least engaging and not bringing enough information to the scene. These results support the hypothesis that \( \alpha \) effectively modulates collaborative dialogue engagingness in human-machine interaction.

### Related Work

Collaborative dialogue between humans and machines has been proposed as a grand challenge in artificial intelligence (Mathewson and Mirowski 2017a; Martin, Harrison, and Riedl 2016; Brown 2008). Previous methods have used hard coded rules, decision trees, and event representations to generate novel narrative chains (Martin and others 2017). We use a deep neural network-based generative language model enhanced with universe model information in the context of improvised theatre (Mathewson and Mirowski 2017b).

While neural response generation systems provide a trainable end-to-end system for language generation, these methods are prone to providing generic, unspecific responses (Li and others 2015). Advances have improved generated responses by optimizing sentence encoding and decoding jointly, post-generation candidate re-scoring (Bordes, Boureau, and Weston 2016; Vinyals and Le 2015; Sordoni and others 2015), reinforcement learning (Li and others 2016a), hierarchical models for distilling extended context (Serban and others 2016), and auxiliary training objectives, such as maximizing mutual information (Li and others 2015), and personality specificity and consistency (Li...
Our work is related to the controlled generation of text using disentangled latent representations (Hu and others 2017). Previous work has used a topic-transition generative adversarial network to enforce smoothness of transition of subsequent utterances (Liang and others 2017). These methods use neural encoder-decoders and generate responses given an input sequence and a desired target class for the response.

Other work has aimed to improve candidates returned by retrieval-based conversation models (Weston, Dinan, and Miller 2018). These methods utilize a conversation model to find similar prototypes using embedding distances and refine prototypes with a sequence-to-sequence model (Guu and others 2017). We do not refine candidates from the conversation model, rather we sample and select using a scoring function defined by the revealing and concealing parameter.

Similar to universe models, topic models or lexical fields have been shown capable of tracking general subjects of a text (Blei, Ng, and Jordan 2003). Dynamic topic models characterize the evolution of topics over a set of documents over time (Blei and Lafferty 2006). Our work differs in that we generate dialogue using the evolving probabilistic belief during a single conversation, as opposed to tracking topical shifts over longer time-scales. Using a probabilistic classifier for narrative tracking has been explored previously (Mohammad 2011; Reagan and others 2016). These works used sentiment classifiers to track emotion and plots arcs through narratives. We extend these works by using probabilistic universe models collaborative dialogue generation. While our work uses separate language and universe models, ongoing research aims to steer or control the properties of text generated with language models (Radford and others 2019) using various attribute models during training (Dathathri and others 2019; Saleh and others 2019).

**Discussion and Summary**

While innovations have improved the linguistic quality, semantic alignment, and consistency of utterances generated by neural models, generated conversations still lack interestingness and engagingness. Our work selects engaging utterances by shaping the underlying narrative arc as opposed to improving the training of generative language models. The methods presented are agnostic to both the universe and the conversational model used. Using rules from improvised theatre, we quantitatively define the evolution of interesting and engaging dialogue.

In this work we focus on genre, emoji, and topic-based universe models. Other universe models to be explored involve causality of events, directions of relationships, or audience reaction prediction. While this work explores the interaction between a base conversation model and a universe model, this method could be compatible with image or video generation.

The main contribution of this work is the computational formalization of the narrative arc, an information-theoretic framework for collaborative dialogue interaction. The framework fills a gap in previous research by connecting the utterance-level improvements of language models with the conversation-level improvements of universe tracking. This is done by sampling candidates from a conversational model using a universe model and the narrative arc. We illustrate narrative arcs underlying popular dialogues and show how universe models can be combined with conversation models to aid in interesting dialogue generation. We present empirical results showing how the narrative arc can improve accuracy on a next line prediction task. Finally, we present an expert user-study to validate our model.

**References**


Figure 5: Varying $\alpha$ for (left) top-3 accuracy, (right) mean reciprocal rank in universe model modulated prediction task.
Five C’s for Human–Computer Co-Creativity —
An Update on Classical Creativity Perspectives

Anna Kantosalo and Tapio Takala
Aalto University School of Science
P.O.Box 15400, FI-00076 AALTO
anna.kantosalo@aalto.fi, tta@cs.hut.fi

Abstract
This paper presents a domain independent framework for discussing human–computer co-creativity. It expands on Rhodes’ (1961) four perspectives on creativity and their later adaptations to socio-cultural views of creativity and computational creativity. The new framework allows the attribution of creativity not only to individual creators but to a collective of creators, recognising the importance of meta-level communication to the creative collaboration, and the variety of creative contributions that emerge during a co-creative process. It also elaborates on the different communities and contexts surrounding co-creative collaboration and thus facilitates the analysis, evaluation and study of human–computer co-creativity by allowing researchers to describe and situate their work in the field.

Introduction
Human–computer co-creativity is a sub-field of computational creativity, which considers collaborative creativity between at least one human and at least one computational agent. This collaborative activity has been defined as mutually influential contributions (Davis, 2013), mixing of human and computational initiatives (Yannakakis, Liapis, and Alexopoulos, 2014) and the sharing of creative responsibility (Kantosalo et al., 2014). In recent years we have seen many relevant practical contributions to human–computer co-creativity emerge in various domains, including e.g. visual arts (Davis et al., 2014), poetry (Kantosalo et al., 2014), game content generation (Yannakakis, Liapis, and Alexopoulos, 2014), and music (Bown, 2018). However, apart from a few models intended for describing human–computer co-creativity as a process (see e.g. Davis et al., 2015; Kantosalo and Toivonen, 2016; Hoffmann, 2016), the fundamental, domain independent factors characterising human–computer co-creativity have received little attention.

Meanwhile, human creativity researchers (Rhodes, 1961; Gläaveanu, 2013) and computational creativity researchers (Jordanous, 2016; Lamb, Brown, and Clarke, 2018; Corneli, 2016) have suggested that to gain a thorough view of creativity it should be viewed from multiple perspectives. Process is just one of these perspectives, which usually include also the creative individual, their creative products and the society (Rhodes, 1961; Gläaveanu, 2013; Jordanous, 2016) and material (Gläaveanu, 2013) context surrounding the creative activity. An earlier attempt to use some of these perspectives to analyse human–computer co-creativity was made by Kantosalo (2019) in her thesis. However these traditional perspectives have been designed for individual creativity, and despite loosely incorporating the ideas of social creativity (Gläaveanu, 2013), or the possibility of computational or human creative individuals and their interactions (Jordanous, 2016), they are insufficient for examining human–computer co-creativity, which may deal with varying numbers of participants, complex processes mixing human and computational initiative, and a myriad of contributions that take place before arriving at a final product.

In this position paper we first examine Rhodes’ (1961) original 4’Ps of creativity framework, Gläaveanu’s (2013) elaborations and Jordanous’ (2016) translation of it for the field of computational creativity. We then present our new framework for human–computer co-creativity, which describes human–computer co-creativity as the interactions within a human–computer collective, the collective’s collaboration process and creative contributions to a community, all situated within a rich context. We then move on to describing communications within the framework and finally proceed to discuss how the framework could be used for describing current systems and the design and evaluation of future co-creative systems.

Classical Perspectives on Creativity
As the interest in creativity as a psychological phenomenon surged in the 1950’s (see e.g. Plucker, 2001), defining creativity itself became a topical task. Rhodes (1961) participated in this discussion suggesting that instead of a single uniform definition for creativity different definitions of creativity together offered four interwoven perspectives on creativity: the person, the process, the product and the press. Rhodes’ framework has remained relevant over time acting as a way for researchers to position their own research within the field of creativity (Gläaveanu, 2013). The framework has also been popular within computational creativity (see e.g. Corneli, 2016; Jordanous, 2016; Lamb, Brown, and Clarke, 2018) and an early attempt to use it for describing human–computer co-creativity was made by Kantosalo (2019). In this section we first examine Rhodes’ original
framework, then visit Glâveanu’s (2013) extension of it and finally describe how the framework has been used in computational creativity research.

Rhodes’ Four Perspectives on Creativity

Rhodes’ (1961) four perspectives are based on 40 definitions of creativity and 16 definitions of imagination. His analysis concluded that creativity is typically used to describe only a part of a multifaceted phenomenon, which includes aspects of the creative person, the creative process, the creative product and the person’s relations to their environment, the press. Together the perspectives are known as the 4 P’s.

To Rhodes (1961) the person perspective considers properties of the creative individual and their relation with creativity. He focuses on identifying creative persons and considers the effects of their physical and mental abilities on creativity.

The process perspective in Rhodes’ (1961) original formulation examines the mental processes of idea creation. Rhodes focuses on identifying the stages of creative processes, and what motivates the process. In addition he is interested in how the creative process can be taught and how it differs from problem solving.

Rhodes (1961) uses the term press to describe the relationship between humans and their environment. According to him the influence of this press can take multiple forms: It can foster creativity during adolescence, or affect the mental processes of an individual during a creative process. He is interested in measuring both aspects of the environment as well as how a person reacts to them.

Finally Rhodes (1961) defines the creative product as an idea or concept produced in tangible form. He focuses on analysing and categorising ideas and differentiates between new concepts and innovations, which he considers as improvements to existing ideas. He considers products can also be categorised according to use, media of expression, utility or aesthetics.

These perspectives give a comprehensive overview of the creativity of individuals. However, although Rhodes (1961) admits that great inventions are not the work of a single mind, he does not describe creative collaborations or elaborate how collaboration is reflected in the different perspectives. The framework is therefore not ideal for describing co-creativity.

Glâveanu’s Five Perspectives on Creativity

Glâveanu (2013) criticises Rhodes’ (1961) 4P’s framework for a focus on the individual and a lack of connections between the perspectives. After a short review of other extensions to the 4P’s, he thoroughly updates the framework to better suit the modern focus of creativity research on social and cultural aspects of creativity.

Glâveanu’s (2013) approach draws from embodied and distributed cognition, where mental processes do not only occur inside a person’s brain, but are situated and distributed between the person and their environment. He is also inspired by “distributed creativity”, an area of creativity research focused on the social factors related to creativity. Reflecting these theories he redefines Rhodes’ (1961) perspectives as actor, action, artifact, audience and affordances.

Glâveanu’s (2013) actor is a person, who exists in a wider social community. The actor perspective expands the person perspective to consider not only the individual traits of the creative person, but in what kind of roles and how the actor performs in their social context. The interactions between the actor and their social context are bi-directional: the actor can both affect and be affected by their context as they work within it or in coordination with their peers.

Glâveanu’s (2013) action perspective attempts to capture both the psychological processes of creativity, as well as their external, behavioural manifestations. These actions are also situated in a context. The action perspective considers the creative process in a broad sense, incorporating both physical actions, such as painting a line, as well as the related perceptual processes.

Glâveanu’s (2013) artifact is again a wider interpretation of the product perspective. Glâveanu considers products to be often seen as separate from the creative person and the process. His artifact is a rich object characterised by both contextual interpretations and meanings as well as its material properties, if it has any.

Glâveanu (2013) divides Rhodes’ (1961) press perspective into two complementing perspectives: the social aspects of the press are represented by the audience, while the material aspects of the press are represented by the affordances. Glâveanu considers that during a creative act an actor is in interaction with multiple audiences involved in the emergence of the new artifacts. The affordances perspective then again considers the material constraints and supports of creative action.

While Glâveanu’s (2013) framework gives more merit to the social interactions a creative person has with their environment and their audience during their creative process, his framework still does not consider collaborative creative activities within a group of artistic peers, limiting the applicability of the model to human–computer co-creativity.

Use of Perspectives in Computational Creativity

Literature

Rhodes’ (1961) original perspectives have also been used by computational creativity researchers: Jordanous (2016) has applied them to computational creativity and discussed their use in the evaluation of novelty and value. Her application of the framework has been used by Corneli (2016) to analyse design principles for creativity and by Kantosalo (2019) in an early attempt to describe different perspectives to human–computer co-creativity.

Jordanous (2016) considers the producer to be a more appropriate term for a creative computer. According to Jordanous, the producer has both physical and functional characteristics: The functional characteristics are described by the characteristics of the creative system, including for example its abilities to demonstrate skill, imagination and appreciation, while the physical characteristics are described by the embodiment of the system in hardware. Alternatively Jordanous proposes that the producer in computational creativity could also refer to the human collaborator in co-
creative scenarios, or individuals involved in the development of the system.

According to Jordanous (2016), the process could consider specific algorithms employed by a system, or interactions between multiple systems, humans or the environment. For Jordanous the product in computational creativity is very similar to the product in Rhodes’ (1961) original framework. She considers that producing good products has so far been one of the most successful areas of computational creativity research. Her interpretation of the press perspective considers mainly the area of social creativity research and bias against computational creativity.

Jordanous’ (2016) re-formulation of Rhodes’ (1961) framework already considers some aspects of co-creativity, allowing the producer role to be taken up by multiple creators at a time, and considering the interpretation of the creative process as interactions between different producers. However her work does not consider these aspects in detail and does not reflect upon what co-creativity means to the product and press perspectives.

Lamb et al. (2018) have written a review of computational creativity evaluation using Rhodes’ (1961) original four perspectives. They selected to use the original 4P’s and not Jordanous’ (2016) adapted perspectives in order to comply with psychological literature. They do not therefore extend the original framework. However they note that for evaluation, the press perspective is an important construct for considering questions related to who is evaluating and the cultural context of evaluation affecting e.g. whether a product is perceived as P- or H-creative.

Kantosalo (2019) has used Jordanous’ (2016) version of the framework to describe human–computer co-creativity. While her approach offers an interesting discussion of co-creativity in connection with traditional creativity research, the four perspectives alone do not offer an independent description of co-creativity, which is one of the goals of our new framework.

Five Perspectives for Human–Computer Co-Creativity

By reflecting on Rhodes’ (1961) 4P’s, Jordanous’ (2016) adaptation of them and Glèveanu’s (2013) 5A’s we have derived the following descriptive definition for human–computer co-creativity:

The creative human–computer collective consists of at least one human and one computational collaborator. The collaboration of the collective consists of individual and collaborative creative processes and interactions that support them. The collaboration results in an artefact or a product that represents the contributions of the collective. These contributions are communicated to and shared with a wider community of peers, audiences, and other social influences. The co-creative collaboration takes place in a context representing the environment of the creative act, including e.g. cultural artefacts and conventions, and more immediate factors such as material affordances and shared mental resources, such as the creative task.

Together the highlighted terms collective, collaboration, contributions, community and context form a new framework that is designed for discussing human–computer co-creativity from different perspectives. We elaborate on different parts of the framework below.

Collective

A collective is formed by the entities actively involved in the co-creative collaboration. In human–computer co-creativity a collective always consists of at least one human and one computational collaborator. Thus the collective perspective distinguishes co-creativity from individual creativity through the number of active creative individuals.

The term ‘collective’ was chosen to evoke parallels with an ‘artist collective’; a group of artists interested in working together on a specific topic. In Rhodes’ (1961) terms the human collaborator could be called a ‘person’, and in Jordanous’ (2016) terms the computational collaborator a ‘producer’, while Glèveanu’s (2013) neutral term ‘actor’ fits both. We prefer the word ‘collaborator’ for the individuals in the collective, as it has been used previously in human–computer co-creativity literature (see e.g. Guckelsberger et al., 2016; McCormack and d’Inverno, 2016).

The collective forms a single unit within human–computer co-creativity that allows for separating the actively participating artists from the surrounding community and context. The collaborators within the collective have a direct say in the internal working methods, goals and artistic processes of the collective, while the influence of the surrounding community and context on the collective’s work is less direct and often filtered through the individual collaborators.

The interactions within the collective can be different from interactions with entities outside of the collective: Interactions within the collective are guided by the dynamics between the collaborators. These dynamics can consist of shared goals and history, or the preconceptions, assumptions and other mental models the collaborators have constructed about each other. Interactions with individuals outside the collective can be twofold: they can happen one-on-one between any individuals within and outside of the collective or the collective may also choose to interact as one entity with the rest of the community and the context. When interacting as a single unit, the collective may for example choose to present one single framing for its creative outputs.

The creativity of the human collaborator has been a strong focus of traditional creativity literature. Studies suggest that the individual creativity of the human collaborator is affected at least by task motivation, domain knowledge, and creative thinking skills (Amabile, 1988), while creative collaboration between humans is affected e.g. by how well the creative partners complement each other, interpersonal facility, gender and age (Abra, 1994). The relationship between individual creativity and collaboration is dynamic and collaboration can also change the individual (Amabile, 1988).

In human–computer co-creativity the human collaborator is often approached as a member of a user group (e.g. Davis et al., 2014; Kantosalo et al., 2014; Kantosalo, 2019, p.13). Few studies have been conducted to examine how individual
properties of the human collaborators affect co-creativity, but initial studies suggest that their user experiences are affected at least by their expertise (Clark et al., 2018).

As Jordanou’s (2016) analysis shows, the producer perspective in computational creativity has also examined the methods and capabilities of computationally creative systems. Most of her examples focus on systems that are somewhat autonomously creative. Intentionality and limited self-awareness have also been suggested as qualities for computational collaborators (Davis et al., 2015), however co-creative systems may also be less autonomous than the original computational creativity methods they are based on (Kantosalo et al., 2014). Therefore autonomous systems may not offer a perfect comparison for current computational collaborators.

Reflecting the human collaborator, the computational collaborator may also have other properties, besides its creative capacity, which affect the work of the collective. These include, for example, what the computational collaborator is capable of communicating and how, as well as its representation, which can be either embodied or a software interface (Kantosalo, 2019, p.15).

Like Gláveanu (2013) suggests for his actor perspective, the collective perspective can also be used for analysing the roles of the human and the computational collaborator. So far several roles have been suggested for the computational collaborator in the collective. These include e.g. functional roles such as support, enhance and generator (Maher, 2012), behavioural roles, such as pleasing and provoking agents (Kantosalo and Toivonen, 2016) and older roles stemming from creativity support system literature (see e.g. Lubart, 2005; Nakakoji, 2006).

Collaboration

Collaboration within the collective considers the individual creative processes of the collaborators and how these are fitted together to form a collective creative process. It also includes meta-level interactions, such as agreeing on common goals, exchanging information and discussing working methods, which are typical also for non-creative human–computer collaboration (see e.g. Terveen, 1995). This new collaboration perspective is broader than the original process perspective. It implements both Gláveanu’s (2013) ideas on mental and physical behaviours, as well as Jordanou’s (2016) suggestion of adding interactions.

In human-human co-creativity, the organisation of creative work can take many forms. Abra (1994) has suggested four dichotomies for co-creativity between humans: fixed vs. on-going, intimate vs. remote, horizontal vs. hierarchical, and homogeneous vs. heterogeneous. Following Kantosalo (2019, p.16), we adopt these also for describing collaboration in human–computer co-creativity, however, we also add a fifth dichotomy, human-initiative vs. computational-initiative to further describe the collaboration dynamics between a human and a computational collaborator.

Abra’s (1994) first dichotomy, fixed vs. on-going considers time: A collaborative process can have a fixed deadline or extend over a longer time. Human–computer co-creativity research has mostly focused on short term laboratory experiments, with the exception of a few long-term musical metacreativity collaborations (Kantosalo, 2019, p.16).

Abra’s (1994) second dichotomy, intimate vs. remote considers co-located and remote collaboration. Collaboration with a non-embodied agent may resemble remote collaboration due to limited communication (Kantosalo, 2019, p.17).

Abra’s (1994) third dichotomy, horizontal vs. hierarchical examines the organisation of the creative process. In horizontal collaboration the collaborators have equal decision making power, while in hierarchical collaboration dominance and power considerations are introduced into the collaboration. Many current co-creative systems introduce a hierarchical process, in which the human collaborators input is given priority over the computational collaborator’s input (Kantosalo, 2019, p.17). A hierarchical relationship may even be the preference of the human collaborators (see e.g. d’Inverno and McCormack (2015)).

Abra’s (1994) final dichotomy, homogeneous vs. heterogeneous considers the distribution of different tasks among the collaborators. In homogeneous collaboration both collaborators work on similar tasks, while in heterogeneous collaboration the tasks are different. This idea is also explored in human–computer co-creativity literature (see e.g. Yannakakis, Liapis, and Alexopoulos, 2014; Kantosalo and Toivonen, 2016).

We add to these a fifth dichotomy, specific to human–computer co-creativity, which expands on Abra’s (1994) horizontal vs. hierarchical dichotomy: the human-initiative vs. computational-initiative dichotomy. This dichotomy characterises the dynamics of the human–computer co-creative collaboration through initiative. Initiative has been discussed extensively in human–computer co-creativity literature: Yannakakis, Liapis, and Alexopoulos (2014) define human–computer co-creativity through the mixed human and computational initiatives. Clark et al. (2018) argue that co-creative interaction can be described either as pulling (human initiated), pushing (computer initiated) or both. Karimi et al. (2018) consider that this spectrum from human-initiative to computer-initiative dominion affects a multitude of factors from frequency of communication to what is communicated.

In addition to the dichotomies described above, human–computer co-creative collaboration can also be described as a series of actions through which it proceeds. In such a series, the individual creative processes of the collaborators are often fitted together by following a specific form of interaction. Typical strategies include for example turn taking, in which the human and the computational collaborator take explicit turns in assuming an active role (see e.g. Winston and Magerko, 2017).

In essence the collaboration aspect adds to individual creativity the need to discuss common goals and context, and the organisation of work. The different dichotomies offer different options for arranging the collaboration, which in turn may or may not affect the individual processes of the collaborators. Nevertheless in a successful creative collaboration the collective benefits from the profound communication and sharing of contributions within the collective.
Contributions

A creative process usually results in a creative product. However, in the field of human–computer co-creativity the human and the computational collaborator typically exchange artefacts or parts of them already during the co-creative collaboration (see e.g. Davis et al., 2014; Kantosalo et al., 2014; Clark et al., 2018). We think that these bits and pieces are vitally important in co-creativity as they form a part of the interaction and facilitate the collaboration in the collective. We call these complete and incomplete creative artifacts contributions. Like Rhodes’ (1961) products and Glaveanu’s (2013) artifacts, contributions can also be immaterial. We call physical contributions tangible contributions and immaterial contributions intangible contributions. Intangible contributions cover all meaningful inputs into the creative collaboration, such as the evaluations and feedback provided by the collaborators during the collaboration. If a distinction between contributions during the co-creative process and the end product is needed, we suggest the term ‘final contribution’ to describe the latter.

As Jordanous (2016) describes, computational creativity has been quite successful in delivering quality products in different fields. The contributions of co-creative collaborators also take multiple forms across various domains. In practise the contributions of the collaborators can be different in both quality and quantity (see e.g. Yannakakis, Liapis, and Alexopoulos, 2014), which is also true for human–human co-creativity (Abra, 1994).

Qualitatively different contributions in human–computer co-creativity could include for example the following:

- An inspiring artefact from the same or a different domain
- A suggestion for an artefact or part of it
- An evaluation of an artefact or part of it

In principle the contributions perspective could also be applied to non-creative collaboration. However, in co-creativity, we are usually interested in contributions that are themselves evaluated high by traditional computational creativity product evaluation measures, such as novelty, quality, typicality (Ritchie, 2007), or surprise (Grace and Maher, 2014). However, to gain best results for the overall collaboration, collaborators should not necessarily use universal standards for these metrics, but adjust their evaluations according to their collaborators instead (Grace et al., 2017).

While the evaluation of different contributions or final artefacts generated by the collective can utilise similar measures as the evaluation of the creative product in computational creativity, assessing or quantifying the contributions of different collaborators has proved more difficult. This is in part because the contributions of different collaborators may not manifest in a visible way in an end product (Kantosalo, Toivanen, and Toivonen, 2015).

Community

According to Jordanous’ (2016) the press perspective in computational creativity is focused on the social aspects, including bias against computational creativity, however there are additional contextual factors that affect the design of co-creative systems (Kantosalo, 2019, p.19). Following Glaveanu’s (2013) ideas we have divided Rhodes’ (1961) press perspective into the social and material environment. In our framework the social environment is represented by the community perspective. This community may include additional artistic peers, audiences, critics, curators, collectors and other individuals and institutions outside the collective, who may also present biases towards the collective.

In Glaveanu’s (2013) framework the audience may be interpreted to represent some aspects of co-creativity through its potential contributions to the creative work. We think it is more useful for co-creativity to separate the collaborative collective from individuals, who are interested in the contributions of the collective and may interact with the collective, while still remaining outside of it. There is also evidence to support that humans view the same co-creative systems differently when they are acting as an audience and when they are actively collaborating with the system (Bown, 2015b), suggesting this distinction is important in practise as well.

Context

Following Glaveanu’s (2013) ideas the context in our framework represents the material surroundings of the work, while the social influencers of the environment are represented by the community. Following Csikszentmihalyi’s (1988) influential idea about situating creative work in a creative domain, we have integrated also cultural aspects of the environment into the context. Our context thus includes the materials with which the collective interacts during co-creation, the previous influential works the collective may draw inspiration from, and the cultural norms and rules which may affect the collective and its work.

Our perspective encompasses Glaveanu’s (2013) view of the material environment influencing or even participating in the creative work, an idea also shared by some computational creativity scholars, such as Bown (2015a). But it also incorporates two core concepts of computational creativity; the inspiring set and external knowledge bases (see e.g. Ventina, 2017; Ritchie, 2007). This implicates our context as a rich surrounding for the co-creative collective, which thus interacts and affects the work of the collective in many ways. As such, the context becomes an important perspective for designing co-creative experiences, similar to the design context in interaction design (Kantosalo, 2019, p.19).

Communication in the Framework

Together the collective, collaboration, contributions, community and context perspectives paint an interconnected picture of human–computer co-creativity. There are many connections between the different perspectives of the framework, which can be partly described through the complex communications that can occur before, during or after co-creation. While communication can be seen as a major part of collaboration, two other parts of the framework, contributions and context, have a special role in facilitating it. As depicted in Figure 1, contributions facilitate communication as a medium, and the context acts as a background for it.

Contributions have an important role as a communication medium within the framework. Different contributions can
be used to establish common ground within the collective. The tangible contributions act as collective aids for embodied and distributed cognition, reflecting the theories that inspired Glăveanu (2013) to separate the material and social context from each other. The intangible contributions, such as evaluations also mediate and direct the communication within the collective during the collaboration and link different contributions to each other. Thus, following Ritchie's (2007) views on computational creativity evaluation, the different chains of contributions could also be used as 'evidence' of the creative behaviour of the collective.

The context provides a background for communication in the framework. It can constrain or support communication within or between the different perspectives. The context can be interpreted in two ways: as the idealised objective context describing the real world accurately, and to the individual subjective context that are incomplete interpretations of the ideal context. The ideal context includes all societal norms and material affordances which constrain and support the work of the collective. It can also act as a bank of inspiration, including well respected masterworks from different fields. As such it has similar properties to Csikszentmihalyi’s (1988) domain, which can also connect artists to each other via contributions.

In order to be able to work effectively in a context, the individuals in the collective (and community), need to negotiate and share their subjective interpretations of the contexts with each other. This may include for example negotiating different time constraints or the use of materials. Through this negotiation the collective may form a more accurate view of the objective context. This may allow the individuals in the collective to gain access to materials they would not have been able to use alone.

Negotiating a shared understanding of the context is a meta-level task, and like other meta-level communications, it is best viewed through multiple perspectives at a time. In addition to viewing how the collective communicates about collaboration, the meta-level perspective can be used for example to view how members of the collective may gradually change as a result of the collaboration (Abra, 1994; Ter-veen, 1995), or how the collective could be influenced by the community and its aesthetic through commission of works. Viewing these communications on the meta-level through various perspectives can also be utilised to give the collective meta-creative capabilities for reflecting and controlling its own work (see Linkola et al., 2017).

**Discussion**

There are several ways in which the framework could be used in practise. These include the description and comparison of human–computer co-creative systems, analysing and planning evaluations, as well as planning new research.

The framework allows for describing and comparing systems on different levels of detail, identifying different aspects important for co-creativity: The collective perspective allows us to recognise and differentiate different co-creative systems by the number and properties of participants. For example the collective of the Drawing Apprentice system (Davis et al., 2015) includes one human and one computational collaborator, while the collective of the Curious Whispers system (Saunders et al., 2010) consists of three computational collaborators and one human. This perspective can also be used to examine different aspects of the collaborators: the computational collaborators in Curious Whispers are embodied, while the Drawing Apprentice is a software based collaborator. It can also be used to analyse the relationships between collaborators in different settings.

Through the collaboration perspective we can investigate questions related to the organisation of work within the collective. For example Clark et al. (2018) compare two different ways to arrange work with linguistically creative computational collaborators, one in which human collaborators are limited to work on one sentence at a time, only receiving input from the computational colleague after returning their contribution, and another with which human collaborators can request further contributions from the computational collaborator at will.

Through the contributions we can attempt to estimate what happened during the collaboration or discuss the authorship of the final contributions (see e.g. Kantosalo, Toivanen, and Toivonen, 2015). The context and community perspectives also allow us to identify different domains of work.

Karimi et al. (2018) argue that who evaluates and what to evaluate are important questions for co-creativity evaluation. This view is echoed by Lamb, Brown, and Clarke (2018) for computational creativity. Jordanous (2016) argues that computational creativity evaluation should consider multiple perspectives. Following her approach, we recommend using the different perspectives for discussing what to evaluate. But we also consider they can be used to discuss who conducts the evaluation.

For what to evaluate Karimi et al. (2018) suggest four targets: the outcomes of the collaboration, the creative process, the creativity of the user, or the interactions between the user and the system. These correspond to the contributions, collaboration, and collective perspectives in the framework, which considers the interactions as part of both the collective and contribution perspectives. However, as Lamb, Brown,
and Clarke (2018) suggest the community perspective allows also for assessing the bias in evaluation. The context perspective then again allows for assessing the effects of material surroundings to co-creativity, following Bown’s (2015a) ideas about the role of materials.

For discussing who evaluates creativity in co-creativity, Karimi et al. suggest three potential evaluators: “the AI, the user and a third party” (Karimi et al., 2018, p. 105). These correspond to the computational and the human collaborator and the community perspective in our framework. Similarly Agres, Forth, and Wiggins (2016) have considered internal and external evaluation of musical metacreation, reflecting a distinction between evaluations done within the collective to improve its work and evaluations received from the community. The added benefit of our framework is that it makes it possible to discuss the relationship different potential evaluators have to each other and the evaluated perspective.

The framework could also help researchers to design their systems and allow them to define how different parts of their system interact with each other. This can be used to select interesting research questions. For example, researchers might deliberately examine different ways of organising collaboration keeping the collective, contributions, community and context perspectives equal.

Finally, by combining different perspectives we may begin to analyse the complex societal role of co-creativity. This includes how the contributions of the collective may break or change societal norms (Shneiderman, 2000), or how a collective may use its contributions to harm a community, e.g. by creating fake news (see Bown and Brown (2018)).

Conclusions

We have presented a new framework for viewing human–computer co-creativity from five perspectives named the collective, the collaboration, the contribution, the community and the context. The suggested perspectives have been inspired by Rhodes’ (1961), Gläveau (2013) and Jordanous (2016). To incorporate different aspects of co-creativity the new perspectives are more extensive than the perspectives suggested in prior frameworks.

The first three perspectives, collective, collaboration and contribution can be used to distinguish co-creativity from individual creativity by the number of participants, through the identification of integrated creative processes and meta-processes related to organising creative work in a group, and by acknowledging contributions to the creative artefact that can include partial artefacts or useful evaluations and feedback. The collective and context perspectives offer a way to situate co-creativity in a wider socio-cultural and physical setting, while offering a way to analyse the effect that individuals and materials outside the collective may have on co-creativity.

Acknowledgments

This research has been funded by the Academy of Finland (decision #311090, Digital Aura). We thank Prof. Hannu Toivonen for useful discussions on Rhodes’ 4P’s and Henkka Hyppönen for making us aware of Gläveau’s 5A’s.

References

Bown, O. 2015b. Player responses to a live algorithm: Conceptualising computational creativity without recourse to human comparisons? In Proceedings of the Sixth International Conference on Computational Creativity, Park City, Utah, 126–133.
Davis, N. 2013. Human-computer co-creativity: Blending human and computational creativity. In Doctoral Consortium of the Ninth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 9–12. AAAI.


A Speculative Exploration of the Role of Dialogue in Human-Computer Co-creation

Oliver Bown
Interactive Media Lab
University of New South Wales
o.bown@unsw.edu.au

Kazjon Grace
Design Lab
University of Sydney
kazjon.grace@sydney.edu.au

Liam Bray
Interactive Media Lab
University of New South Wales
liam.bray@student.unsw.edu.au

Dan Ventura
Computer Science Department
Brigham Young University
ventura@cs.byu.edu

Abstract

In this paper we consider the notion of dialogic creative artificial intelligence (DCAI) systems, where co-creativity between a human user and a computational system is supported through dialogic interaction. By dialogue we mean both traditional language-based communication for explicit critique and persuasion (dialogue about artefacts) as well as a broader potentially non-linguistic notion of dialogue that emerges through the exchange of suggestions and changes (dialogue through creative artefacts). To capture this, we define DCAI as occurring when both system and user are able to influence each others’ creative objectives. The paper motivates our pursuit of dialogic interaction and provides some explanation of why we have defined it through its impacts rather than its mechanism of action. We provide two analyses to support our argument: an exploration of a commercial creativity support tool that has an extensive vocabulary for describing artefacts abstractly but does not meet our definition of dialogic interaction, and a case study of a creative interaction between two human professionals that exhibits dialogic interaction throughout. For the latter we consider how studies of human-human co-creation can offer non-obvious design concepts that might be applied to co-creative DCAI systems.

Introduction

The use of computational creativity algorithms to create applied usable co-creative systems is now a significant focus of research in the field, requiring a combination of computer science, design and creative practice-focused research methods. Dialogue, both overt linguistic discussion and non-linguistic exchange of concepts and artefacts, is a critical component of human creative collaboration. In this paper we explore how different ideas of ‘dialogic interaction’ might be realised in co-creative systems, both in addition to and instead of a more rigid graphical mode of human-computer interaction (Shneiderman 1983). The main purpose of this paper is to develop an understanding of the nature of dialogue as an interaction design concept, what design requirements it presents and what challenges it poses for the current state of the art.

We do this by considering, as small case studies, one existing piece of commercial music generation software, and one observational study of a commercial music production team working together in a collaborative creative process (without the use of AI tools). These small case studies are used to examine the gap between existing modes of human-computer interaction involving creative AI systems, and potential scenarios that may more closely resemble human-to-human co-creative activities. As such, this is viewed as speculative design research (Auger 2013), albeit without a design artefact. We attempt to abstract the features of both types of interaction such that we might consider a range of design possibilities (i.e., not necessarily fixating on imitating the human-to-human scenario).

We begin by outlining some expectations of what dialogic interaction might entail in creative contexts, laying the foundation for the development of co-creative agents exhibiting what we call dialogic creative artificial intelligence (DCAI). We then go on to evaluate an existing piece of generative music technology according to these expectations, considering how it falls short, but also how it comes close in some ways, to satisfying the requirements for DCAI. Following that, we present our observations of a real human-to-human co-creative activity that clearly exhibits dialogic interaction. Lastly, we piece these elements together to consider some within-reach possibilities for supporting DCAI.

A Vision for Dialogic Creative AI

Co-creativity, in the context of computational creativity research, describes a scenario in which a human and a computer collaboratively make contributions to a creative outcome. It is common to frame a vision of co-creativity with reference to human-to-human forms of collaborative creative production, where the computer becomes promoted from a “mere” tool to a creative partner (McCormack 2008). For example, Lubart discusses four human-to-human metaphors that help categorise how computers might become creative partners: the computer as nanny, the computer as pen-pal, the computer as coach, and the computer as colleague (Lubart 2005), each describing a different way that the computer makes contributions to the output. In some cases, the computer’s output could clearly be called creative in its own right with its own original contributions, whereas in others it is focused on giving stimulating feedback to the human user, to drive their creativity further. This may involve critique (i.e. communicated justified evaluation), which is by some accounts at least as challenging as, and important to, the creative process as generation.
The rise of commercially successful conversational interfaces has drawn attention to the potential for more fluid forms of interaction with creative AI systems that may support creativity in new ways, including co-creative interaction that better resembles human-to-human co-creation. In comparison to the potential of conversational interaction, our current predominant model of human-computer interaction for computational creativity is one in which we trigger a ‘generate’ action on a traditional GUI interface and some output is generated. We may then iterate the generation process, tweaking parameters and selecting presets or data sets to refine our search, but essentially still engaging in a graphical instruction-based interaction modality, where a GUI provides the means to operate a given system. It is easy to see why this makes it hard to consider the system as a particularly active agent in the co-creation process.

Relative to this, a ‘new’ conversationally-grounded model of interaction would step away from the limits of the GUI interface and employ the open-ended nature of conversational interaction to support something more closely resembling human-to-human co-creativity. This vision needs to be unpacked, as current state-of-the-art conversational interfaces are often very constrained and do not guarantee to offer greater open-endedness of interaction. It is not their conversational interface per se that would afford greater creative potential, but the capability of the underlying computational creativity technology, for which any number of interface designs might be suitable to exploit this capability. As an aside, we speculate that the second property (adaptability) may be critical in establishing the first (co-influence): a system can only actively and positively influence a creative process through proper engagement with the human’s changing objectives. Again, the notion of ‘loose coupling’ can be useful in thinking of how two actors combine into a single temporary creative agency in the formation of co-creative outcomes. All of this serves to say that our vision for future co-creative systems is one where they have greater co-influence on the creative process, and that in doing so the interactions we have with them can be considered more dialogic than tool-like.

**Motivations for Designing for Dialogic Creative AI**

Our focus on dialogic systems is motivated in part by the emerging interest in understandable AI, and in part by our understanding of creative processes. Understandable AI is important because people are known to be more inclined to engage with faulty or less-effective AI systems if they receive direct, understandable responses from them, when compared to highly optimised systems that are neither directly responsive nor understandable (Oudah et al. 2018).

A lack of comprehension is known to lead to mistrust and disuse of systems (Hancock et al. 2011), which in a creative context we think means little to no adoption of our systems beyond our own tight community.

Elaborating further on what it might mean to productively influence a creative process, we note that creativity research has long examined the need for divergent, heuristic, search-based cognitive work. This is often by definition: many consider creative tasks as those for which a clear path to the outcome is not known, and some form of blind (or at least ambiguity-tolerant) search must be employed (e.g., Simonton 2011; Perkins 1996; Sternberg and Lubart 1991; Amabile 1996). In human-human co-creativity, the technique of ‘brainstorming’ epitomises this type of divergent search in the form of a formalised activity. Supporting ‘expansive’ search is one way in which a co-creativity tool...
might be expected to positively influence an outcome, and we already have examples of such work in the literature. For example, Karimi et al. demonstrate the use of a co-creative sketching tool that aims to actively stimulate a greater diversity of outcomes, and hence greater ‘creativity’, achieved through the stimulation of novelty, through its own sketch suggestions (2019).

However, most of the above-cited models of the creative process also stress that divergent search is, in itself, insufficient; there must be a process of review and filtering that tests ideas against goals and may drive the rethinking of those goals. Various models of creativity also specifically address the need for convergence, either in terms of convergent forces that constrain and direct divergent search, or as a separate phase. Simonton discusses the sketches used by Picasso, leading to his masterpiece Guernica, which demonstrate the presence of divergent blind search (2007). If nothing else, convergence is implied in the fact that creative tasks result in specific outcomes, but some tasks might involve gradual refinement. A dialogic co-creative process could simply stimulate divergence and go no further, but we might define it as more complete if it follows through to the process of narrowing in on a final outcome.

This also suggests that our two properties are related through the idea of the co-creative partners being loosely coupled; convergence can be achieved through the dual actions of the computer influencing outcomes but also adapting in a skilful, context-aware way, the success of which would be in part due to its ability to support convergence where relevant.

For convergent, focused creative development in particular, but also for expansive divergent development to a lesser extent, we believe that the system must be good at adapting to context and be capable of explaining itself (or giving context to its actions). We note that as well as helping discover a solution that fits preconceived criteria, expansive search can also involve arbitrariness, which helps frame future creative goals and set helpful, narrowing constraints. It can be good to ‘just get something on the table’. For example, Stokes’ research into creative practice shows how artists set themselves up with an individual style through the establishment of constraints, which can in theory be arbitrary, as long as they serve to isolate the artist’s individual style (2009).

Situating Dialogue in Different Interaction Paradigms

As we note above, we do not think DCAI is necessarily bound to conversational interaction (i.e., natural language), but we do recognise that there is a natural fit here. We believe dialogue could be achieved in a range of interaction contexts as a way to exchange information about creative artefacts without necessarily exchanging information through them. One natural situation for a dialogic interaction would be a request-based scenario (Bown and Brown 2018), where the user requests the system to produce outputs (“give me a funky bassline”) and then iterates in response to what is offered (“make it more syncopated”, “more like that last bit”). This requires some shared understanding of the subject, which requires the system both to adapt its understanding and to communicate it. A good explainable system might provide context or even be persuasive: “what about something in the style of Stanley Clarke?”, or “a more syncopated rhythm would make sense against this drum beat”.

We also note that a more ‘ambient’ (Bown and Brown 2018) mode of interaction might also provide a solid foundation for dialogic co-creativity. The bassline could simply be playing along in its own track while the producer works on something else, adapting to other elements in the music, without much direct communication, but with contextual factors influencing how the bassline develops.

These different modes, and the different levels of initiative, encapsulation and agency they represent are important to consider in terms of the overall user-experience and acceptance of the system. Clippy, the notorious Microsoft Paperclip, was disliked by many users for its intrusiveness and misplaced confidence, but many of the services it offered are actually automated as standard in today’s word processors (Maedche et al. 2016). Under the right circumstances the user’s sense of the system’s involvement and engagement might enhance the experience rather than be irritating or counter-productive. Such design considerations can help frame how forms of dialogic interaction can be useful in co-creativity.

Analysis of an AI Music Production Tool

With this notion of DCAI in mind we turn to consider an existing commercial AI music production tool in terms of how these concepts play out in a provisionally co-creative scenario. We have not attempted to set this up as a formal user study because, to our knowledge, the tool is yet to be adopted in real production scenarios, and we feel that this is necessary for a formal user study to be relevant. Instead we take a form of heuristic walk through the properties of the system in a similar vein to types of heuristic evaluation used in interaction design research (Nielsen 1995). The aim is both to provide a worked example of how co-creative agents can be more dialogic in their interactions, as well as to motivate why that would be a good idea.

The tool is Splash Pro, the first offering of an end-user tool from Australian Music AI startup Popgun. Splash Pro is a generative creativity support tool aimed at amateur music producers, which runs as a standalone piece of software for Mac and Windows and also as a plugin for a number of music production environments. It works in the symbolic music domain, for which MIDI is the standard. We examined the tool running as a plugin for Ableton Live. This analysis refers mainly to the design of the interface, not to the generative capability of the system, although we recognise that the two are intertwined.

Splash Pro has three generation modes—create, accompany, and blend—with three different choices of instrument: keys, bass and drums (vocals is also an instrument option, but only in ‘create’ mode). Create mode produces a single melodic line or chord sequence, bassline or drum beat, with options to choose a style, which varies according to the instrumentation chosen. Accompany mode produces an accompanying chord sequence, bassline or drum beat, given
a source sequence to accompany. Blend mode generates an
original sequence, but does so as a musically valid interpo-
lation of two existing sequences, selected by the user.

In all cases, the interface requires the choice of a musical
key and a duration over which to generate (it may also make
use of the music’s time-signature, which is not an option in
the Splash interface, but can instead be accessed by plugins
from the host program, where it is a user-settable parameter.
Ditto with tempo). It also offers the user the choice to input a
four-chord progression over the given number of bars.

Further, for each combination of generation mode and in-
strument, the system offers a number of preset stylistic op-
tions, as well as a number of numeric parameters. For create
mode, with bass, preset options include either genre-tags:
electronic, hip-hop, pop, RnB and rock, with each genre
having subcategories such as hip-hop:boombap and hip-
-hop:modern. Alternatively you can choose a “technique”
instead of a genre: long notes, short notes, on-the-beat short
notes, accent on beats 1 & 3, busy moving notes, plays to-
towards end of bar, short notes with groove, assortment of
grooves, funky groove, less funky groove, unstable funk, etc.
In addition, a set of numeric parameters can be modified:
variation, note length, root octave, velocity, swing, and den-
sity. For other instruments, the options are varied to reflect
that instrument. For example, for the numeric parameters
for keys, chordiness, timing and pitch range are added, and
swing and density are removed. This gives Splash Pro and
its users a strong and shared vocabulary for reasoning about
musical artefacts, which we see as a starting point for dia-
logic interaction.

Splash Pro has a graphical user interface based largely
on buttons, popup menus and sliders, meaning that it has a
defined set of GUI operations and options, rather than al-
lowing any kind of open-ended construction of structures
or behaviours by the user—an “operation-based” interaction
paradigm (Bown and Brown 2018), as opposed to a request-
based or ambient paradigm. As such, it presents as a regular
software tool rather than a creative assistant.

It would be easy to conclude that such a tool does not
satisfy our definition of dialogic interaction. However, as a
generative software tool, we note that the system is implic-
itly involved in autonomously making ‘suggestions’ (even
though its outputs may not be conceived of as such). It is
widely documented how such output, even if not gathered
in a dialogic manner, can strongly influence creative outputs
when used by creative practitioners. Thus, in order to con-
sideer a minimal form of dialogic interaction, it is interesting
to consider what would be needed to have the system adap-
tively respond, and complete a loop of loose coupling.

One option for extending Splash Pro would be enabling
the user to give feedback to the system, such as ratings that
might feed back into a generative process as an error sig-
mal (for example with reinforcement learning or evolution-
ary computing). This would permit adaptivity in both di-
rections: the system to actively influence the user, and the
user to actively influence the system. Alternatively, it could
be possible for the system to ‘ambiently’ adapt and respond
(which may not be evident through the GUI): even if the
user doesn’t give feedback, the system may be able to glean
information through their actions.

Without some form of these adaptive behaviours Splash
Pro, despite its extensive descriptive vocabulary, would
seem to fall short of our criteria for being a dialogic sys-
tem. This is no slight on the system: we are not convinced
any successful instances of this exist in current creative soft-
ware. Yet we might go even further in admitting Splash Pro
as a dialogic system, since it provides a means for a user to
feed it existing musical input, and through this input, it actu-
ally has a much more complicated interface than what is
visible in the GUI. To fully examine this interface requires
considering how someone might feed input into it.

Furthermore, through iterative processes, the user might
fill in the gaps in what might make up a true dialogic interac-
tion. For example, if the user wants to explore the interaction
between different styles, they can combinatorially explore
this space by feeding outputs back into the system. They
could use the blend tool to blend two outputs that come from
the create tool, and they could also use the accompany tool
iteratively through different styles to create original combi-
nations. Our informal experiments with Splash Pro, as well
as with similar tools such as Google Magenta’s Ableton Live
plugin suite, have indicated that iterations of the generation
process, feeding back content into the system’s input, could
constitute what might be called a pseudo-dialogic process.
We assume the system is not adapting its internal state, but
from a philosophical standpoint, when the system is coupled
with the user and the musical content being worked on, the
musical input can be thought of as an external state.

Although there is no option for feedback and no evidence
of ambient adaptation in Splash Pro, we can still consider
how it affords a sense of dialogic interaction through the
user’s own exploration of its response to different inputs and
getting to know its behaviour. In the simplest sense, this is
afforded through the sense of suggestion given in the stylis-
tic parameters, as well as the sense of adaptation as the user
themselves adapts and gets to know the behaviour of each
of the generation settings. Since these are likely to be very
complex models with rich behaviour, this sense of adapta-
tion may be quite significant, even if no actual adaptation is
happening.

Thus a rich enough system that was not adaptive could
still present a sense of dialogic interaction by enabling the
user to systematically iterate and progress ideas through
their use of the system, with the shared creative workspace
playing the role of a pseudo-adaptive intermediary: this may
be the case if the system has very rich behaviour or a rich
space of options, combined with some means to explore it.
Exploration by interaction here can include submitting dif-
f erent types of inputs to the system experimentally. Even
simple pop-up lists of options might be structured to create
rich interactive pathways through a co-creative process. The
system may have a large enough set of ‘canned’ responses to
user inputs that navigating its interface still gives the expe-
rience of adaptation and suggestion. These would not meet
our criteria for being dialogic, but they may increase a user’s
sense of the agency of the system.

A well-furnished vocabulary of terms shared between
user and agent is, we argue, not sufficient to make a system
dialogic—for that it needs to be adaptive as well as actively and positively influencing the user. But a dialogic interface could be built on top of the current Splash Pro UI to allow the user to explore co-creation through the composition of operations; composition here referring to the creative and possibly adaptive iteration of these operations. Thus the user could say, “could it have a bit more of a Latin feel?’ and the system could potentially present two or three different ways in which that adaptation to user input could be offered, with explanatory output to support that. The user could give positive or negative feedback on any such suggestions, driving the system to adapt how it blends these processes, and the system could offer more persuasive suggestions, drawing on previous success.

A Case Study of a Professional Music Production Process

We now turn to considering a real human-to-human co-creative scenario, studied through observation. Since dialogic creative AI systems—at least as we’re envisaging them—don’t exist, we set out to investigate a collaborative creative session between two musicians in order to identify how their interaction was dialogic. We are not suggesting by doing so that interaction with DCAI systems should imitate human-to-human interaction. Instead, we seek inspiration from human-to-human interaction to seed design ideas and requirements.

Our subjects, Uncanny Valley (UV), are a commercial music production studio based in Sydney. They are responsible for producing cues for major television dramas, reality TV, news and sport titles in Australia. They are also collaborative partners in our research into usable computational creativity systems and interested in adopting creative AI techniques in their workflow.

We observed two UV artists in a collaborative composition session: a producer, composer and keyboard player (henceforth A), and a co-composer and guitarist (henceforth B). Their brief was to create an alternative version of a track already previously composed by them, and used for TV coverage of a popular sports show. The alternative version was being used to promote a specific event within the upcoming season. The main melody of the original composition was to be kept and the track reworked given a specific reference track from the popular music canon (“The Man” by The Killers) which was provided by the client. B elaborates that the aim of such reference music is not to “sound alike” but to “vibe alike” to this reference, with a focus on tempo, groove and sound (a “big” sound in this case). This is a recurring topic for the team who are often given reference tracks for playing to (not to be used in the final version), which expressed the groove; A emphasising how critical it was to establish the groove right from the start. Here “groove” refers to the basic rhythmic expression and emphasis.

A elaborates on the use of exploratory search in the early stages of a track: “Writing a track from scratch is sometimes a different process, but it will always start with a jam, either on piano or guitar. Before we did [another track] recently, it started with a played piano riff and then we wrote the melody over the piano riff. Then after the song was pretty much written as an entire thing, that’s when we started producing it. Whereas this is more producing, writing at the same time. I like to produce and write at the same time; it saves time.”

A refers to the initial jams as “expression sessions”. B elaborates: “I’m self-confessed not a master of any instrument, just sort of spittingballing at the wall, isn’t it? Mud at the wall” B adds: “I’ll leave the room almost deliberately often so that I can come back with observational power, otherwise sitting here listening to everything; I’ll come back and go ‘Oh, it’s great, but that’s the hook!’ It’ll be like ‘Really?"
I thought this was the hook!' like ‘No, that’s the hook!’ Because I can hear it from downstairs and it’s been going around and around and I can hear people whistling it downstairs, whatever.” This highlights the iterative, negotiated re-framing of the creative task: B even deliberately removes themselves from the room in order to force the adoption of a different perspective. Here, dialogue that influences the creative objective is not only something that happens, but something that is being actively sought.

A also elaborates further on the need for live variation: “if you’ve got that little error there or little timing thing or that little note that you play, little ghost note that you play that you wouldn’t do by yourself then you’ve got an extra layer of goodness there to add into the final mix. That’s what gives music it’s x-factor sometimes a little bit ... I’ve just learned that over the years of producing, if there’s enough of that random stuff, jammy stuff going on at the beginning, then at the end you’ll have something a bit better.”

Reflecting this, the duo play together when recording parts (A performing the main melody on a MIDI keyboard), keeping eye contact, talking and using hand signals whilst playing. This is described as being done to get the groove, as well as being efficient. There’s an interesting dichotomy in the way they talk about efficiency: in the early stages of an expression session, novelty, divergence and “jammy stuff” are clearly valued, but as the process continues, the focus shifts to speed and predictability, often after a specific meta-discussion of the need to progress. We suspect this is a common component of professional creativity: iteration and exploration are necessary, but deadlines are always tight.

A says: “At the early compositional stage of a track and finding all the bits and pieces that work together, if every part is aware of every other part then you keep... I think the reason I was jamming with B is because he can play the guitar and I can’t, and I wanted to find what keyboard part was going to work over the guitar part, and then once we’d recorded that we sort of went, what bass part’s going to work with this? And with the groove. And often it’ll be happy accidents with jamming as well.”

As well as using expansive, ambiguity-tolerant and diverse search processes to develop themes, the team also elaborate on how this process feeds their ideas about the overall structure of the track. It is apparent that determining the structure depends on specific content elements, and that this is strongly interdependent—the specifics of the elements in turn depend on where they sit.

A says: “Sometimes it’s the case of thinking about what we’re going to produce in the future while we’re writing the riffs to understand what we need to get… That’s cool.” Thus as they jam and isolate guitar parts they like, they collaboratively develop a concept for the overall structure. This is described as “groove 1, groove 2, chorus” at one point, but later there is agreement that the second section is a variant of the first: “Groove 1, groove 1 on steroids, and then chorus”.

As the order of parts is locked down, the specific details of the guitar riffs are also rapidly honed. A small handful of iterations of each take ensues, sometimes full repeats, clearly converging on specific phrases. The transition between exploration and exploitation is fast, and often explicitly encouraged by B, who was expressing a desire for the session to be as short as possible.

Having laid down the main part, the focus was turned to more specific details, such as “turnaround” moments, variations on the main themes emphasising the transition from one section to another, which involve discussion of the context of the transition (from where to where).

Multiple parts involve more extended discussion between A and B, taking different positions on what should be done. One such question was whether the parts should play more in unison or against each other. Through this back-and-forth, the two musicians attempt to actively and positively influence each other, and this dialogue is occurring both through music and discussion: it is both communication-through-creation and communication-about-creation.

Discussion

The human-human example clearly satisfies our definition of dialogic interaction, beyond the simple fact that the participants are in conversation. B inputs content that is informed and influences A, but also adapts to A. The team are very quick to converge on a stylistic focus, which combines their musical expertise with their experience of working together. The team moves beyond issues of genre and style rapidly to focus on finer specifics. Once within this stylistic focus, there continues to be a lot of rapid interaction which takes place through talking, gesturing and playing. It is fast and exploratory. We would describe their search phase as exhibiting a rapidly iterated curve of expansion followed by convergence. We note that during the expansive phase, many factors are up in the air: the song structure has still yet to be decided, and part of the convergence involves fixing the context for which the parts are being made. The collaborators navigate this ambiguity by establishing a shared but abstract dialogue about the desired qualities of the music: they don’t need to debate extensively what a “big” sound is, they are both familiar with each others’ stylistic vocabulary, and being too precise too early would harm the exploration.

We condense this to three observations regarding how a generative system might better support a dialogic co-creative process in a commercial music composition context, which may carry to other areas of creative practice.

Firstly, **interdependence**: the compositional process involved a great deal of interdependence (“all parts of the track must talk to each other”). The overall track structure depended on the components that arose and vice versa; it was beneficial to have parts played together to capture the same groove, which was established early on, and to carefully consider what elements might move in tight unison.

Secondly, **agreement of purpose**: the use of a stylistic reference was made in very specific ways. Through their experience and shared understanding of the process, the team was able to move very quickly into a very specific and targeted ideation stage. There was substantial ambiguity in their representation of this purpose, which was not just tolerated but encouraged by both parties.

And thirdly, **search processes**: the team made early use of fast-moving, expansive, exploratory (yet very stylistically-focused) search, laying out lots of options, guided by rapid
dialogue. They invited happy accidents and were attentive to quick re-conceptualisations of their goals and expectations, for example stepping out of the room. There was also very rapid but necessary convergence through refinement, once a number of basic decisions had been made. These convergences were often explicitly discussed, a kind of metadiscourse about the process that shows another critical role for dialogic interaction.

We hypothesise that making our co-creative systems more dialogic will help create that sense of shared narrative that is known to characterise great human collaboration (Gottschall 2012). This is supported by our observational study—the musicians quickly developed a shared, abstract description for their piece that was more story than tight categorisation. People exchange information, feelings, and intentions using narratives (Polletta et al. 2011; Bruner 2009), and our proposed dialogic approach to co-creativity seeks to leverage this idea. Recent preliminary work in the domain of human-machine teaming validates this approach by showing that when AI systems are equipped with a rudimentary form of narrative intelligence, they are more likely both to establish and to maintain cooperative long-term relationships with people (Crandall et al. 2018; Oudah et al. 2018; Goodrich et al. 2018).

Narrative impacts people’s view of robots (Rosenthal-von der Pütten, Straßmann, and Mara 2017), including perceptions of their usefulness and behavioral intentions. For example, Mara et al. observed that introducing robots as fictional characters increased people’s perceptions of a robot’s usefulness and behavioral intentions (2013). When thinking about what will make co-creative systems more likely to be adopted beyond their own creators, building trust and perceived utility through narrative seems promising. As narrative is a field of computational creativity in its own right, it may even be possible to explicitly represent and reason about this shared narrative that emerges from dialogic interaction, using ideas like co-operative narrative generation (Pérez y Pérez 2015). Computational narrative might also offer some ideas about how to evaluate dialogues (Kypridemou and Michael 2014; Rowe et al. 2009; Wang, Chen, and Li 2017), an essential part of dialogic AI that we’ve not touched on at all.

Dialogic interaction also has clear overlap with the notion of explainable creative systems. End-user explainable AI (as opposed to explainable techniques aimed at engineers and researchers) is in its infancy, but as it develops it will quickly become a critical component of co-creativity. Without the ability to explain suggestions, it will be impossible for a system to persuade, critique, or justify its behaviour, all of which would help build trust and acceptance (Zhu et al. 2018). Explainability has even been proposed as a possible “meta-aesthetic”, or fundamental drive, for creative agents, based on the notion that ineffable ideas—those that cannot be expressed, or at least cannot be expressed concisely—are ineffective (Bodily and Ventura 2018).

The notion we are trying to capture with DCAI—the idea that co-creativity interaction should be about a shared dialogue through and about the work—is not a new one in AI. Enactive AI proposes a focus on autonomy and adaptability through interaction (Froese and Ziemke 2009), based on the notion of enactive cognition. Enactive cognition posits that reasoning arises through intentional interaction with the environment, and places a great emphasis on communication, both linguistic and non-linguistic (Cuffari, Di Paolo, and Jaegher 2014). This has been discussed in the field of creativity research as “distributed creativity” ( Sawyer and Dezutter 2009), the idea, reminiscent of Csikszentmihalyi’s systems view (2014), that collaborative creativity is “non-individualistic”, and “emerges from the improvised dialogues of the group”. Enactive cognition provides support for our notion that dialogues can occur through interaction, and therefore need not necessarily imply chatbots or voice interaction, saying that co-operation can arise without “high-level mental processes” like direct communication (Fantasia, Jaegher, and Fasulo 2014).

Conclusion

We have proposed dialogic creative AI, a class of computationally co-creative system that we hypothesise is worthy of further study. DCAI has two core features: both human and artificial agents must have the potential to actively influence the creative objective, and must be able to adapt their behaviour as those objectives change. We have based this definition on the outcomes of establishing dialogue, rather than by defining dialogue itself. This focus exists because we want to capture a broader range of communication than traditional direct exchange of language—we want to capture both dialogue-through-artefacts and as dialogue-about-artefacts. In short: It’s not sufficient to slap an explanatory chatbot on top of your generative model, although that sounds like an excellent starting point. Our definition contends that there must be two-way influence and adaptability. We believe focusing on dialogue construed this way is essential for building creative systems that will be more broadly adopted in the future. Making that a reality will mean grappling with a host of related ideas: explainability, framing, critique, initiative, intent, meta-aesthetics, and more.

Acknowledgements

The informal study of Splash Pro was assisted by Chloe McFadden, a student in the Bachelor of Media Arts, UNSW, acting as a research intern. This work was supported by the Australian Research Council Discovery Project grant DP200101059.

References


Co-Creative Songwriting for Bereavement Support

Lee Cheatley*, Margareta Ackerman+, Alison Pease*, Wendy Moncur*
*University of Dundee, UK {lecheatley, apease, wmoncur}@dundee.ac.uk
+Santa Clara University, Santa Clara CA mackerman@scu.edu

"You are always going to miss someone when they are gone, really, but I think writing about it and especially just being able to hear it played back and then being able to sing along really helped me come to terms with what I had written and how that could be interpreted”
—Study participant (P1)

Abstract
Self-expression is essential to processing our thoughts and feelings and is central to successful mental health therapy. Art therapy provides a wider range of expressive mechanisms than offered through traditional approaches, allowing individuals to process their emotions when traditional therapies prove unsuccessful. Yet, effective expression through art therapy may call on a level of artistic experience that is not available to all. As such, a lack of expertise or comfort with artistic expression may hinder one’s ability to receive needed mental health support.

Creative machines can offer novel therapeutic approaches by offloading the need for creative expertise and opening up creative self-expression to those who lack the corresponding experience. In this paper, we focus on bereavement, and explore a co-creative songwriting system, ALYSIA, as a new form of therapy for those who had recently suffered the loss of a loved one. We evaluate the utility of this creative system in aiding bereaved individuals through several case studies. In addition, we discuss the utility of co-creative systems to the therapeutic context with potential application to a broad range of therapies.

Introduction
Why, and for whom, do we develop computationally creative systems?

The question above will become increasingly important as our community transitions from showing that it is possible to develop computationally creative (CC) systems, to showing that it is useful to do so.¹ The question of why has been

¹Of course, showing that it is possible is by no means an accomplished task: while we may have largely convinced ourselves, many members of society remain to be convinced. Assessments of whether computers can be creative may well change along with changing cultural and sociological perceptions of creativity, perhaps ultimately depending on whether the assessor wants to define “creative” in such a way as to include machine processes and output.

considered in a variety of contexts and domains, such as the argument in (Pease et al. 2019) that research in CC can help to address key challenges in both Automated Reasoning and Automated Scientific Discovery.

The question for whom was partially explored in (Colton et al. 2015), where Colton et al. define creativity stakeholders as “people who may have something to gain or lose from software which is creative” (Ibid., p2), and suggest a non-exhaustive list of “researchers, the wider AI community, funding bodies, experts in the psychology of human creativity, neuroscientists, artists, art critics, journalists, philosophers, educators, the public, and so on” (Ibid., p5).

In this paper, we offer a further answer to the question; suggesting that CC systems can be developed in order to offer novel therapeutic approaches by offloading the need for creative expertise and opening up creative self-expression to those who may lack the corresponding experience. We take as inspiration the proposal in last year’s International Conference on Computational Creativity (Cheatley, Moncur, and Pease 2019) – that CC systems can be applied to therapeutic fields – to suggest a new purpose and a new type of stakeholder. Here, we investigate one of the key provisional design recommendations for CC bereavement support tools in (Cheatley, Moncur, and Pease 2019), to Require users participate in the creation process, where this may “support users in interacting with their grief and lead to the creation of meaningful possessions.” (Ibid., p4).

Creativity can play at least two roles in a therapeutic context: a created artefact, such as a collage of photos of someone who has died, and the process of the bereaved person putting together the collage, can both be very meaningful in the grieving process. These two roles co-align with the twin strands of research in CC: autonomous creativity in which a system creates an artefact, and co-creativity, in which system and person work together. In order to investigate the design requirement above, we focus here on co-creative systems. In general, co-creative systems are developed to benefit experts, although there has been some work in developing co-creative systems for novices, such as (Compton and Mateas 2015).

To explore the design requirement to Require users participate in the creation process, we consider how co-creative systems can be applied or developed to enable novices in a particular domain to have deep and meaningful interac-

Proceedings of the 11th International Conference on Computational Creativity (ICCC’20)
ISBN: 978-989-54160-2-8

33
tions in a sensitive context to engage in bereavement experience. We start by reviewing the literature on grief and bereavement, bereavement technology, CC, and the co-creative songwriting system ALYSIA. We then present our threepart study, in which we perform in-depth user testing on three recently bereaved participants, where we firstly introduce participants to ALYSIA through a series of videos; secondly observe them using the system to create a song related to their bereavement; and thirdly conduct semi-structured interviews in order to explore their experience in using ALYSIA in a bereavement context. In our Results section we elaborate on three key themes that emerged from our interview data: (1) Supporting self-expression; (2) Therapeutic value; and (3) Receptiveness to ALYSIA and songs created. Finally, we conclude by discussing our findings and suggesting general lessons for the CC community in this new context.

**Background**

Grief is one of the most challenging experiences that most of us face during our lifetime. The passing of a loved one can increase the risk of mortality and carry significant risks to mental and physical health, including increased rates of heart attacks, strokes (Carey et al. 2014), increased blood pressure (Buckley et al. 2012), and aggravation of physical pain (Bradbeer et al. 2003). If participation in bereavement and grief is undertaken with little care, or avoided completely, these ill effects can amplify. Effectively coping with grief often necessitates the ability to engage with it, while a lack of engagement can prolong and complicate the process (Worden 2018).

While essential to the success of therapy, many find it difficult to express and engage with their grief: Patients may feel reluctant to express themselves in front of others, find themselves incapable of doing so verbally, and fear judgement (Ryder and Hospice 2018).

Helping the bereaved express themselves is a central goal of therapy. Mental health practitioners seek to provide a safe space in which people feel comfortable doing so (Rogers 1957). Some people, however, struggle to put their thoughts and feelings into words. This contributed to the rise of arts therapies, which have proven successful in helping some people express themselves and engage with their feelings (Lord 2018).

**Grief and bereavement interventions**

Grief includes the process of adapting to a world without the deceased whilst maintaining a place for them in their life (Klass, Silverman, and Nickman 2014; Worden 2018). The bereaved often oscillates between avoiding and interacting with their grief, both of which can be beneficial (Schut 1999).

Formal interventions usually take the form of one-on-one sessions or group therapy with a mental healthcare professional. Interventions traditionally focus on person centred therapy or counselling in which the bereaved express themselves verbally (Arnason 2001; Newsom et al. 2017). Less traditional, person-centred arts therapies offer the bereaved an opportunity to express their grief through a creative medium (Lord 2018; McClocklin and Lengelle 2018; Dalton and Krout 2005).

Most traditional bereavement interventions are not based on technology, but rather traditional talk therapy. When technology is used, systems designed to support those going through bereavement focus less on creation and more on curation and social interaction. For example, people make memorial pages on websites such as Facebook and write to or about the person they have lost. These interactions can be helpful to the bereaved (Refslund Christensen and Sandvik 2015; Refslund Christensen and Götved 2015; Christensen et al. 2017). However, the public nature of these methods may also put the bereaved in the presence of malicious users who can cause them emotional distress (Christensen et al. 2017; Phillips 2015; Sabra 2017).

Many systems have been theorised and implemented on a small scale by HCI researchers, mostly focusing on the bereaved person curating digital possessions to be placed within a physical container (Banks, Kirk, and Sellen 2012) or accessing already curated possessions (Kirk, Reeves, and Durrant 2011; Odom et al. 2014). Similar work, seeks to include the bereaved in the creation process of the memorial object (Story Shell) itself (Moncur et al. 2015). They found that the participant felt they had benefited therapeutically from recording memories to be incorporated into Story Shell. (Moncur et al. 2015) report that the therapeutic benefits were a result of the participant feeling that they had a receptive audience in the researchers, and could also be a result of continuing bonds with the deceased - the participant mentioned they found themselves addressing the deceased in some recordings. Despite this, Moncur et al and the participant found that other people were reluctant to contribute recordings for Story Shell as they were unsure what to record and wary of the recordings being shared. This suggests that systems, much like therapies, which facilitate user participation in the creation of memorial objects could be helpful for the bereaved, while emphasising the importance of being able to use such systems privately.

**Art therapy interventions**

Interventions have been successfully applied in a variety of contexts, with more traditional forms benefiting from the therapeutic relationship and exploration of thoughts and feelings (Worden 2018). Nevertheless, the effectiveness of formal interventions is debated (Jordan and Neimeyer 2003), and further they are not always accessible when and where needed (Ryder and Hospice 2018) and can be costly.

Art therapy encourages patients to engage in the bereavement experience by expressing and exploring their thoughts and feelings, and has been shown to open up the benefits of traditional therapies to those more able to express themselves creatively (McClocklin and Lengelle 2018; Moss 2010; Glover et al. 2016; Lichtenenthal and Cruess 2010). Art therapists often offer a series of workshops to their clients, holding off on more creative activities until they have had time to build a therapeutic relationship with the client and make them feel comfortable being creative (Moss 2010; O’Connor et al. 2003; Kohut 2011).
Arts therapies have been reported to enable people to express themselves, gain new insight, make sense of their loss and continue bonds with the deceased. (McClocklin and Lengelle 2018) spoke specifically about the benefits of writing as part of recovery from grief. They argued for several advantages of writing over talking: (1) it can be done privately; (2) thoughts and feelings can be captured when experienced and shared when the bereaved is ready; (3) the internal dialogue fostered by writing can make it easier to express thoughts or feelings to others; and (4) thinking and writing about bereavement can help normalize the experience, which again can make it easier to talk about.

Music therapy
Music therapy in a bereavement context involves creative songwriting in a clinical music therapy setting and “point[s] to positive growth in bereaved adolescents through creative songwriting” (Dalton and Krout 2005). Songs that emerge from this process are “often emotional, challenging, and deeply thought provoking, and can provide a valuable contribution to our understanding of the experience of terminal illness, death and loss.” (Heath and Lings 2012) The latter of these studies explores the potential of music therapy for a somewhat experienced lyricist and a novice.

Dalton and Krout (Dalton and Krout 2006) conducted a more in-depth study investigating the use of music therapy for bereavement groups. The groups would go through the entire songwriting process together, from theme selection, to writing original drum tracks, melodies, and lyrics, to the performance and recording of the song. They found that this process “proved to be engaging and offered a safe, creative method of addressing the difficult subject matter of a loved one’s death.”

Dalton and Krout argued the “structured flexibility” of their methodology “allowed group members to creatively address the five grief process areas and discuss individual issues related to their loved one’s death” and the lyrics created by participants “showed insight and creativity in identifying, expressing, and processing personal issues related to areas of understanding, feeling, remembering, integrating, and growing.”

These works suggest that the creation process, and the exploration of the created work, are as important, if not more so, than the final product.

Computational Creativity Systems for Bereavement
Cheatley et al. argued that CC can usefully be applied to therapeutic fields (Cheatley, Moncur, and Pease 2019). They investigated reminiscence practices of 13 bereaved participants, exploring possessions used to support reminiscence and participants’ receptiveness to CC being used in this context. They used their findings to identify the following 10 provisional design recommendations for CC in a bereavement context: (1) Be available freely online; (2) Output physical and digital possessions; (3) Present framing information; (4) Incorporate degradation into digital output; (5) Require users participate in creation process; (6) Allow for a varied source of input; (7) Employ sentiment analysis; (8) Allow for and foster repeated use; (9) Allow private and collaborative creation; and (10) Be secure and private. Of these, some (for instance, 1, 8, and 10), are straightforward requirements for system developers. Many, such as (3), require further investigation in order to be thoughtfully designed.

We deem (5) to be one of the most fundamental requirements, since previous work conducted by Moncur et al. (Moncur et al. 2015) and Dalton and Krout (Dalton and Krout 2006) found participation in the creation process can be therapeutic and as such lends credence to this line of inquiry. In this paper we focus on further exploring this particular requirement, while our case study and findings also overlap with some of the other requirements (such as 10).

One of the challenges identified in (Cheatley, Moncur, and Pease 2019) is to encourage people who may not think of themselves as creative to engage in a creative process. Co-creative systems can be applied in the bereavement context to overcome this challenge. Their creative abilities offset or even eliminate the need for any artistic expertise on the part of the bereaved. This open up creative self-expression and the benefits of art therapy forms that are not otherwise accessible.

As an example, songwriting offers significant therapeutic benefit (as discussed above), yet its inherent complexity and multifaceted nature (consisting of lyrics writing, melody composition, music production, singing, etc) makes this form of self-expression inaccessible to most. ALYSIA (Ackerman and Loker 2017) is a co-creative system that removes the barriers that traditionally block most non-musicians from the creation of original songs, allowing everyone to express themselves through this art form.

By offering a flexible co-creative framework that facilitates self-expression, ALYSIA shines a light on what co-creativity can offer to bereavement and potentially other therapeutic practices. Other systems, facilitating expression through, for example, visual art or dance, may offer other forms of healing self-expression. This opens up the possibility of a new direction for co-creative systems: Offering therapeutic value through artistic self-expression to those without the corresponding training and expertise.

In addition to the fundamental offering of a creative partnership that offsets the need for artistic expertise, co-creative systems may carry additional benefits in the form of wide accessibility and cost efficiency.

ALYSIA: Co-creative songwriting
This paper explores the benefits of one co-creative machine partner, the ALYSIA songwriting system, for the bereavement process. In this section we give a brief overview of ALYSIA.

ALYSIA (Automated LYrical SongwrIting Application), has been developed over the past five years (Ackerman and

---

2To enable the novice to write songs, the therapist closely collaborated with the novice over a period of three sessions.

3Physical possessions gradually degrade over time. This refers to simulating a similar process with a digital item.
Loker 2017). Since its inception, the co-creative system aimed to enable everyone, irrespective of their level of musical expertise, to express themselves through songwriting.

The process initiates with the user selecting a backing track in a genre of their choice (choosing from amongst Rock, Pop, R&B, Country, or Jazz). The user subsequently inputs topics to guide the lyrics creation, or chooses topics from a list of common options (‘love’, ‘moody’, ‘anger’, etc).

The first co-creative process involves lyrics creation, which can be initiated by either selecting a lyrical line created by ALYSIA (based on the user’s topic), or having the user input their own lyrical line. ALYSIA generates suggestions for subsequent lines based on previous selections/inputs, adapting to the user’s style if they choose to write any of the lyrical lines. Editing of ALYSIA’s lines is a common use case that allows for deeper engagement.

The next step involves the co-creation of top line melodies, which fit with the lyrics and the underlying musical track. The melody system proposes different ways in which the lyrics can be sang, allowing users to choose from its suggestions, edit them, or input their own. Finally, the user may record the song in their own voice, or choose a male or female singing in-app voice. The voices can also be used to support learning a new vocal song, or to duet with the users.

ALYSIA allows novices to rely heavily on its suggestions, while letting more advanced users input their own lyrics, melodies, or vocals. Users often become more independent through repeated interactions with the system (for example, gradually editing more of ALYSIA’s suggestions, and eventually starting to enter their own ideas for lyrics and melodies). Experienced songwriters find ALYSIA’s suggestions helpful for exploring the creative space and breaking out of writer’s block.

ALYSIA has been commercialised by WaveAI and is broadly accessible to the public on the App Store, allowing its creators to further improve the system through an abundance of implicit and explicit user feedback (Ackerman and Pérez y Pérez 2019). To date, over 30,000 songs have been created with ALYSIA. Users span the expertise spectrum, from novices to established songwriters.

**Experimental Setup**

An evaluative research study (“user testing”) was conducted to explore whether the process of co-creating a song with a computationally creative system can help bereaved people interact with their bereavement and be an enjoyable experience. This involved the researcher introducing participants to ALYSIA through a series of three short videos, and then asking them to create a song related to their bereavement. Participants engaged in semi-structured interviews which sought to elicit further information on: their experience using the system; listening to the song; engagement with bereavement; and their receptiveness to such systems. See Figure 1 for an example depicted on ALYSIA’s melody screen and karaoke screen (the final creation screen).

**Recruitment and Participants** For inclusion participants had to be over the age of 18 and speak English fluently, this was to ensure participants understood what they were asked to do and the potential implications of participation. There were no exclusion criteria set for gender, country of origin, etc. We recruited a total of 3 participants through contacts made in previous studies, snowball sampling. Participants have been anonymised via the assignment of pseudonyms (P1-P3). P1 was a 28 year non-religious female who had been bereaved of her grandmother less than a year ago. P2 was a 57 year old non-religious female who had been bereaved of her mother less than a year ago. P3 was a 56 year old non-religious male who had been bereaved of his spouse 5 - 10 years ago. All three of the participants indicated they were very close with the person they had lost. The perspective of the data gathered, and the subsequent findings have been influenced by these demographics.

**Musical Experience** Of the three participants only one (P3) indicated they had musical experience. P3 commented they play several musical instruments, and had written songs in the past but despite this did not consider themselves an experienced songwriter.

**Ethics and Limitations** The University of Dundee granted ethical approval for the research. We acknowledge several limitations of this initial study: (i) participants did not use the system independently, but in the presence of a researcher, and (ii) the limited number of participants precluded the application of quantitative measures. Future work will include larger samples.

**Analysis** Thematic Analysis (Clarke and Braun 2013) was employed to analyse the responses to the open questions. Data was grouped into themes (coded) and analysed iteratively to refine these themes across all participants. NVivo 12, qualitative analysis software, was used to do this.

**Results**

Three key themes were identified in the interview data on the experience of participants using ALYSIA in a bereavement context: (1) Supporting self-expression; (2) Therapeutic value; (3) Receptiveness to ALYSIA and songs created. These are discussed below.

**Supporting self-expression**

Participants went into the study with pre-conceived notions of whether they were creative in general, and particularly whether they are able to write songs. Subjects P1 and P2, who had no songwriting experience, remarked “I think starting is going to be the hardest part” and “It is quite hard, I’m not very musical.”

Study participants were further concerned about doing justice to the deceased. P1 commented commented that the process felt “daunting because I did not know how to start
it and what to write because I wanted to do it justice.” Similarly, P2 wanted it to sound “nice because you want it to reflect the person, so you don’t want to do a bad job.”

P1 remarked being able to “shuffle through a lot of [lyric] suggestions was really helpful... and it wasn’t showing the same once again which was nice.” P3, used entirely ALYSIA generated lyrics and commented “I thought the lyric writing part gave you a lot more flexibility... instead of asking what you thought it gave you lots of things and you could pick out ones you maybe didn’t realise you thought. That was very clever,” and went on to say: “The lyrics part I thought was interesting because I thought it was going to be really hard to think up lyrics but then there were lots of them there and so that made it really straightforward.”

P1 remarked on the system as a whole, “Writing a song is quite hard, especially if you have never done it before. It isn’t something you think yourself good at. I’ve never been much of a writer, and songs are largely lyrics.” P1 also felt the system gave them: “A lot of control actually, I was surprised. You can choose your melodies and things like that, the background music and you can choose different genres. I think I chose a country song which I don’t think I would have thought that I’d have chosen going into it. It gives you a large variety and you can change that depending on your mood, and you can go back and change things if you change your mind. ... I like the fact that it gave you the option to sing along, and you got to choose the melodies to go with it. ... I enjoyed it. I left feeling better about it, and about my relationship with my grandmother because I was remembering all the good past moments.”

P2 also spoke positively about the system as a whole and commented “This helps towards making people creative because it is like you have got someone else there you are bouncing ideas off.” P2 went on to theorise that in a bereavement context, a creative aid such as this would let you bounce ideas off the system in private “which you probably want to do if you are in a grief situation” rather than doing so with other people. P3, although reportedly constrained by limited backing track options, said “I thought it [creating a song] was going to be hard but it wasn’t. It was really fine. I think it is because I found things that worked for me.”

**Therapeutic value**

Participants reported numerous factors that contributed to ALYSIA’s therapeutic value. Chief amongst these were that the system supports loss-oriented activities such as interacting with feelings, reminiscing, and accepting the reality of their loss, as well as serving as a distraction from the sadness of their loss. Participants also felt that the personal nature of the creative process and the resulting song contributed to the therapeutic value of this approach.

**Engage with Grief**

Participants felt that using ALYSIA helped them engage with their grief, speaking about how creating and listening to their song helped them interact with their feelings, and in some cases discover new dimensions in their grief.

P1 remarked “You are always going to miss someone when they are gone, really, but I think writing about it and especially just being able to hear it played back and then being able to sing along really helped me come to terms with what I had written and how that could be interpreted – how sad it could be and things like that... I guess I was sadder about it than I realise, but not in a bad way.” P1 shared that playing the song back to you “helps you realise what you have written and how you are feeling about it because you...”
are actually hearing it.”

P1 had also commented “sometimes hearing the lyrics back made me feel a bit emotional and sad... It has made me feel kind of better about it. It made me realise how much I miss her but I think it is quite good to remember people that you have lost and to think back on the fond memories, so they are not forgotten.”

P2, like P1, also experienced some sadness creating the song, “I guess it made me feel a bit sad because I was focusing on something I didn’t really want to focus on, I guess”, but “didn’t feel as sad listening to the song.” P2 theorised this was because they “were trying to put [their] feelings into words” which they found difficult to do. P2 felt they hadn’t had time to engage with their bereavement and that they “had to squash it down at the time, because it is not about you, it is about making sure other people are okay. Whereas this is you focusing on you when you are trying to write what did it mean to you, which is harder.” P3 reported they felt the lyrics generated by ALYSIA from which they could select “was helpful”, “very reflective”, and “made me think about things I didn’t really realise I was thinking about because it [ALYSIA] made suggestions.”

Participants also felt using ALYSIA helped them reminisce about the person they have lost. P1 felt ALYSIA helped them “focus more on the positive memories of her” rather than the negatives “which I think is a nice thing to do, especially when the person is gone.” They felt ALYSIA provided them with an opportunity “to think about my grandma and going back on the memories and things you kind of forget about when you’re just living day to day. It was nice. I liked it. There are little parts in the song that bring up other memories.”

P2 shared that “It made me focus a bit more on the good memories and what I enjoyed... it helped me think back on things you would want to be in a song. You want to remember the happy times, or at least I do – I’m sure everyone is different -, but it is quite nice to have that preserved in your own way”. P1 spoke positively about the reminiscence ALYSIA inspired, “I enjoyed it. I left feeling better about it, and about my relationship with my grandma because I was remembering all the good past moments,” and “It made me feel happy”. P1 also shared how they felt using ALYSIA helped them accept the reality of loss alongside their reminiscence, “It was good thinking about her and remembering all the good times and how I’m coming to terms with the fact that she is not around anymore.”

**Personal nature of creative process and creation** Participants felt the personal (or, private) nature of what they expressed to create the song and the setting in which they did so influenced the therapeutic value of ALYSIA and the song, P1 felt singing the song made it seem more personal and therapeutic, but that it led to them being “A little embarrassed I guess, because it is quite personal.”

P1 went on to say, about the song, “I kind of wanted it to be something that people could understand and not too personal... I wanted to keep those things [more personal memories] for me” and “her [the deceased], and my mum, and sister, and grandpa.” P1 didn’t want to “reveal all” about the deceased but wanted the song to be “a little bit personal.”

P2 felt that using ALYSIA in a research setting lessened the system’s therapeutic value “because it is kind of artificial, it is kind of like ripping a band aid off, you are a bit exposed or a bit vulnerable whereas if you were doing it privately by yourself then I think it probably would [be of more therapeutic value].” P3 felt the “element of the audience” make it “less helpful”. P3 felt if they had the ability to sing the song it would maybe have been “a lot more personal, and that would have been good” and felt “music is really personal in the first place”.

**Receptiveness to ALYSIA and songs created**

All of the participants stated they would be open to using ALYSIA again in a bereavement context, and P1 and P2 wanted copies of the songs which they said they may listen to again in the future and share with their families.

In general, all of the participants spoke favourably about ALYSIA and its use in a bereavement context, and some went on to discuss other ways they feel it could help them or others. P1, when asked whether they feel they would use the system again to engage with bereavement responded “I think this does help process what has happened because it makes you think about it a bit more or think about your time with the person or what you are feeling and I think writing them down and expressing them kind of helps you come to terms with it.” P2 responded similarly to this question, “Yeah, to be honest I would and... I think it would be a great thing for kids to actually channel things for them. I think it would be really instructive for them.”

When asked if they were still talking about in a bereavement context P2 responded “For bereavement I think you could use it because it would allow them to explore things maybe that they couldn’t tell you... because I think if you are talking to kids you tend to direct them as to how you think they are feeling which might not be how they are feeling at all and then you could end up making things worse for them because you are leading them down a negative path because that is what you are worried about. Whereas, if you let them play with that you can actually then see the things they are saying and then maybe challenge or channel them in a bit more positive light or get them to think on more positive things.”

P2 also felt a system such as this would be helpful for parents who have lost one child of their children, theorising it would let the adults to “take time out to have your grief yourself... something they could do on their own” whilst they normalised it for their other child(ren). P3 also reported they would use ALYSIA or a system like it again in a bereavement context when asked, responding “Sure. I would use the lyrics part definitely. If I was sitting down to write a song that would be fabulous.”

**Discussion and Conclusions**

In this paper, we explore the potential of a co-creative system for therapeutic purposes. In particular, our focus has been on bereavement, where songwriting was previously shown to help those suffering the loss of a loved one to better process their feelings and express themselves (Dalton and Krout.
Yet, like most arts, songwriting requires skill and practice. To this end, we have selected ALYSIA, a co-creative system designed to help everyone express themselves through song, requiring no musical training or expertise.

Our study explores this therapeutic approach through a case study. This qualitative analysis helps gain insight into the nature of this approach to ascertain what aspects are or are not effective and identify how this therapeutic methodology can be improved. Future work will assess this therapeutic method on a larger number of participants and incorporate quantitative methods, such as the Warwick Edinburgh Mental Wellbeing Scale (Tennant et al. 2007).

Our findings show that participants found therapeutic value in the creative process and the resultant song, even when subjects initially had substantial doubts on their own creative abilities and particularly their abilities to create songs. The use of a co-creative system offers a novel way for the bereaved to engage with their grief and process their emotions. Notably, ALYSIA’s suggestions have allowed participants to discover feelings of which they were not previously aware.

This new form of support for the bereaved has the potential to give a voice to those who otherwise struggle to connect with challenging feelings. As suggested by the participants, this form of support may also benefit children, who may otherwise find it particularly challenging to comprehend and express feelings associated with the loss of a loved one (Segal 1984). Expressive arts have been shown to help children express themselves and cope with grief (Moody and Moody 1991), and as such, expanding the children’s creative capabilities through co-creative systems may be particularly effective for this population.

The ability to support a creative process without the presence of another person may be one of the main benefits of the therapeutic approach proposed here. Professional songwriters often write in groups in order to more effectively and efficiently explore the underlying creative space. On the other hand, the presence of others when creating songs related to bereavement creates discomfort and hinders expression. Future work will let users create songs in the bereavement context privately and in their own time. A creative partner that enables self-expression, without exposing one’s thoughts and feeling to another person, stands to be a new, effective therapy that is uniquely enables through co-creative systems.

Therapeutic computational creativity has the potential to offer the dual benefits of a co-creative process that can be undertaken in private, while also leading to the creation of a piece of art which can validate the user’s experience and feelings and prompt further exploration of those feelings. Participants took pride in their songs (suggesting a sense of ownership for the creative artefact), as well as expressing eagerness to share the artefact with their families.

In this paper, we explore a co-creative CC system (ALYSIA) in a therapeutic context and illustrate its potential therapeutic benefits in a bereavement setting. We believe that CC systems have the potential to provide therapeutic benefit in multiple settings (e.g. depression, anxiety, and mental wellbeing) and to enable diverse forms of self-expression across artistic fields (e.g. visual art, story writing, and poetry).

Co-creative CC systems deployed in a therapeutic context to support bereaved people have the potential to offer widely accessible, affordable therapeutic support. Through interaction with creative machines, users with diverse creative abilities will be able to better connect to themselves and reap the therapeutic benefits of engaging with and expressing their feelings. This promising novel direction widens the scope, application and value of CC, suggesting new reasons and new stakeholders as part of the answer to our question: Why, and for whom, do we develop computationally creative systems?

References


What Happens When a Computer Joins the Group?

Paul Bodily  
Computer Science Department  
Idaho State University  
Pocatello, ID 83209 USA  
bodipaul@isu.edu

Dan Ventura  
Computer Science Department  
Brigham Young University  
Provo, UT 84602 USA  
ventura@cs.byu.edu

Abstract

We consider the advent of computational creativity systems that are potentially far superior to human creators, at least in some domains. To do so, we revisit the concepts of P- and H-creativity and propose a social refinement called group-relative creativity (G-creativity). Using this reconceptualization, we explore several critical questions regarding computational creativity’s effect on human creators.

Introduction

On November 19, 2019, the world was shocked by the sudden and premature retirement of South Korean Go player, Lee Sedol. Sedol, an 18-time world Go champion of 9-dan rank (the highest rank in the game), made headlines in 2016 when he was defeated 4-1 in a match against AlphaGo, a Go playing computer program developed by Google DeepMind. Sedol’s defeat at the “hands” of AlphaGo marked the first time a computer had beaten a 9-dan professional, and it left both Sedol and the rest of the world stunned.

Of his retirement, Sedol said:

*With the debut of AI in Go games, I’ve realized that I’m not at the top even if I become the No. 1 through frantic efforts. Even if I become the No. 1, there is an entity that cannot be defeated. Frankly, I had sensed kind of a defeat even before the start of the matches against AlphaGo. People from Google’s DeepMind Technologies looked very confident from the beginning.* (Yonhap News Agency 2019)

Sedol’s defeat and subsequent retreat from professional Go playing was for many a cause for alarm:

*Sedol’s final bow in professional Go signals a more significant, existential concern. If a world champion, floating at the peak of personal achievement, starts to view human accomplishment and machine accomplishment as one and the same, it creates an environment for frustration, disappointment, and perceived loss of purpose. Sedol sits at the edge of this realization, but all of us are not far behind.* (Pranam 2019)

How did the AlphaGo system designers respond to Sedol’s retirement? DeepMind’s CEO Demis Hassabis credited Lee with showing “true warrior spirit” and then stated:

*On behalf of the whole AlphaGo team at DeepMind, I’d like to congratulate Lee Sedol for his legendary decade at the top of the game, and wish him the very best for the future … I know Lee will be remembered as one of the greatest Go players of his generation.* (Vincent 2019)

Without wishing to lay blame, one cannot help but sense a sort of eulogy in these words, not so much for Lee Sedol as for the era when humans ruled the world of Go. Sedol’s fatalistic retirement and the near-condescending reactions of DeepMind and others betray a sense that we as a society have in some ways already accepted the inevitability of computational domination and at the same time have failed to anticipate and prepare adequately for the consequences of super-human AI and CC systems. And so we pose these questions:

- Do we risk destroying human creativity by creating systems that are more creative than humans?
- Are we as CC researchers doing our due diligence to anticipate the potentially negative impacts that our systems will have on human creativity?
- Are we prepared to take responsibility for these impacts?

Figure 1: Lee Sedol retired as a professional player after being defeated by AlphaGo. *Photo by Google via Getty Images.*
• What can be done to mitigate any negative consequences of CC on human creativity?

Though technological advancement is often a boon for humanity, there are well-known exceptions to this, cases in which such advances are at least correlated with the development of human deficits, displacements from jobs, etc. Factory automation has eliminated many manufacturing jobs; keyboards have eliminated the need for penmanship; GPS means that people don’t learn how to read maps or navigate using waypoints and landmarks; spelling and grammar checkers mean people don’t develop mental models of syntax and grammatical structure; and recent research even suggests that our reliance on the internet for all things information promotes cognitive offloading and may result in negative effects on problem solving, recall and learning abilities (Storm, Stone, and Benjamin 2017).

It has historically been the case that such deficits and displacements disproportionately affect the under-educated. And, while the negative effects are very real for the displaced, they have been largely temporary and transient because they could be compensated for, over time, with additional educational interventions. However, an interesting recent study on the future impact of AI on workers suggests a new trend—for the first time, a major technology (AI) will have the most effect on well-educated, white collar workers (Muro, Whiton, and Maxim 2019). How should we react when education may no longer be the solution? Because CC is still in its infancy and due to the challenging nature of its ultimate goals, one may be tempted to assume that we are not yet facing these issues vis-à-vis creativity; however, we argue that AlphaGo is a CC system—and a dominant one—and that even for CC, these issues are contemporary and unavoidable.

Super-human CC is here, now

Though enjoyed by more than 40 million people worldwide (most in Asia), with a history going back 2,500 years, Go was largely unknown to most of the world until the fateful match between Sedol and AlphaGo. In the game, players take turns placing black or white stones on a 19 × 19 grid to capture opponent’s pieces or to surround empty territories. Go was chosen intentionally as the next great challenge for computational intelligence because despite seeming simple, the game allows for more possible moves than atoms in the known universe, making traditional “brute force” AI methods an impossibility (Pranam 2019). Experts widely believed that Go could not be solved in the way that other games (such as chess) have been.

AlphaGo is a unique example of a computational system that has exceeded human abilities in ways that even unbiased humans readily acknowledge. AlphaGo doesn’t just beat humans, it consistently beats even the very best of human Go players. This phenomenon is especially marked because the problem of playing Go possesses a unique characteristic that many other creative domains lack: given any two Go players, there exists a well-defined and universally-recognized way of comparing them (i.e., which one wins a game they play against each other).

Because this is at least not so clearly the case in many CC domains, it seems appropriate to ask whether Go, in fact, represents a creative domain and therefore whether AlphaGo can, in fact, be considered a computationally creative system.

In order to address this, we can test AlphaGo against two common ways of defining computational creativity. The first, due to Colton and Wiggins (2012) defines CC as

The philosophy, science and engineering of computational systems which, by taking on particular responsibilities, exhibit behaviours that unbiased observers would deem to be creative.

Is playing Go a behavior that is deemed to be creative?

Go is considered an art form in Asian cultures, one that not only allows creativity but also demands it. Consider this from European Go champion Fan Hui:

As we walk the path of improvement, we must study and experience all aspects of the game: joseki, fuseki, shape, and direction, just to name a few. After we absorb this knowledge, we learn over time to apply it flexibly. But to reach the level of grandmasters, even this is not enough! As we gain experience, our knowledge fetters our creativity. To truly throw off these shackles and liberate ourselves from what we have learned, we must discard labels of “right” and “wrong.” In their place, we must consider the essence of Go: the role of each stone, and the relationships between them. Only in this way can we reach the level where invention prevails over tradition. AlphaGo began from the same fundamentals as humans, but the rigid attachment to knowledge is simply not in its nature. (Baker, Hubert, and Graepel 2016)

Or, this from Lee Sedol himself:

It made me question human creativity. When I saw AlphaGo’s moves, I wondered whether the Go moves I have known were the right ones. Its style was different, and it was such an unusual experience that it took time for me to adjust. (Choe 2016)

For those who play Go, creativity is considered a fundamental aspect of the game.

A second common way to define computational creativity is by using common attributes of creativity. Can it be demonstrated that AlphaGo’s choice of moves exhibits novelty, value, surprise, and intentionality—attributes frequently used to characterize computational creativity? Does it generate behavior through computational means that are deemed novel, surprising, valuable, and intentional?

Consider the following analysis of two key points in the match between AlphaGo and Sedol:

In Game Two, the Google machine made a move that no human ever would. And it was beautiful. As the world looked on, the move so perfectly demonstrated the enormously powerful and rather mysterious talents of modern artificial intelligence.

But in Game Four, the human made a move that no machine would ever expect. And it was beautiful too.
Indeed, it was just as beautiful as the move from the Google machine—no less and no more. It showed that although machines are now capable of moments of genius, humans have hardly lost the ability to generate their own transcendent moments. And it seems that in the years to come, as we humans work with these machines, our genius will only grow in tandem with our creations. (Metz 2016)

It seems clear that by widely-accepted standards, the game of Go is considered a creative domain and AlphaGo indeed constitutes a computational creative system. In other words, computational systems that exceed human levels of creativity are already a reality, and we are likely to see an increasing number of such systems in an increasing number of creative domains.

Comparison in Creative Domains

It may be argued that perhaps the creative domain of Go is somewhat unique because it offers a well-defined and universally accepted method for the direct comparison of (the creativity of) individuals. The natural assumption may be that most creative domains do not offer such a comparative mechanism and that thus human creators in most domains may not be as susceptible to the kind of disruptive comparison to which Lee Sedol was exposed. However, we argue that most, if not all, creative domains are subject to the effects of some kind of comparative mechanism, at least implicitly (and for many, it is actually quite explicit), even if they are not as overtly competitive as is the domain of Go.

In other words, people can and do make creative comparisons (or indirect surrogates of such) all the time. Consider things such as which work garners the most viewers/attention/citations, sells for the most money, wins an award (or competition, even!) Given two artifacts from a domain, people can almost always be coerced into choosing which they prefer. Anytime creativity is rewarded/incentivized in a non-uniform way, it becomes, in some sense, an optimization problem, and therefore one of comparison. In some domains, such comparisons may be subtle and even latent and may have little (perceived) measurable affect. However, in many domains these comparisons result in competition for recognition, awards, employment, etc. The result is very often a natural sense of success/failure attached to creative endeavor. Thus, the issue at the heart of Lee Sedol’s “crisis of faith” may soon threaten creators in many other domains because all creative domains include at least an implicit element of comparison and many include an explicit competitive component.

How Should We Think About This?

Computational creativity theory provides the lens through which we typically view the world of CC systems; can it help us make sense of the effect that (dominant) CC systems may have on humans, particularly their creativity? We suggest an advantage in this respect for a more nuanced view.

P-creativity and H-creativity

Directly relevant to the discussion of Lee Sedol is the notion of personal versus historical creativity. Personal or P-creativity represents behaviors or concepts that are novel to their creator, but may not be novel in the broader society. Historical or H-creativity, by contrast, refers to behaviors or concepts that are novel within the broader society (Boden 1992). Though P-creativity is prerequisite to H-creativity, few instances achieve the status of being H-creative. Like many creative professionals, Sedol had devoted his career to the pursuit of H-creativity.

In his announcement last November, Sedol was very specific about his reasons for giving up the hunt for H-creativity. It was not merely that he had been defeated, but that in his estimation, an entity had entered the field that “cannot be defeated”—he believed that H-creativity was no longer a possibility for him or any other human. Is the introduction of a computational agent into a community somehow fundamentally different than the introduction of another human agent? If so, how? What about this scenario is different, say, than when Lee Sedol loses to another really great human player like Lee Chang-ho (the only human player currently ranked higher than Lee Sedol)? Why don’t other human players evoke the same reaction as AlphaGo does?

An apt analogy may be found in Csikszentmihalyi’s (Csikszentmihalyi and Csikszentmihalyi 1992) flow model. In the model, a sense of “flow” (meaning a state of energized focus) is achieved when an individual addresses high levels of challenge with equally high levels of skill (see Figure 2). Certainly H-creativity occurs within these same parameters, and individuals do not remain static within the model. As

![Figure 2: As CC systems improve in their ability to compete with human creativity, humans may begin to feel that the level of challenge for producing novelty and value exceeds humanly-capable skill levels. This scenario, as represented in Csikszentmihalyi’s flow model (Csikszentmihalyi and Csikszentmihalyi 1992), leads to anxiety and doubt, which if left unabated leads to diminished human creativity.](image-url)
skill increases, the perception of challenge decreases such that over time the level of challenge must also rise commensurate with the level of skill in order to maintain the “flow” experience. Of interest in our discussion is what occurs when the level of challenge increases at a rate that the individual’s level of skill is unable to match. In this scenario, the individual tends towards anxiety and worry until either the skill level can be increased or the individual quits the endeavor altogether.

In presenting his framework for CC systems, Wiggins describes a state of limbo that he calls generative uninspiration. In this state, “the technique of the creative agent does not allow it to find valued concepts” (2006). Can it be that computational creativity inadvertently contributes to generative uninspiration in humans? Wiggins’ solution to this problematic scenario is that the system (or in this case, the human) must undergo transformational creativity, or in other words, change its method for generating new artefacts. However, even according to Wiggins, the method for doing so is non-trivial, and for some domains may seem humanly impossible.

**Rethinking P- and H-Creativity**

Because we are interested in creativity in the context of a community whose membership is changing (i.e., by the addition of CC systems), the concepts of P- and H-creativity need some reformulation in order to account for this sociability. To this end, we begin by exploring the relationship between P- and H-creativity.

Let the function \( P_i : D \rightarrow \{0, 1\} \) be an indicator function that maps artefacts from a domain \( D \) to a Boolean value such that \( P_i(x) = 1 \) indicates that an artefact \( x \in D \) is P-creative for an individual \( i \).

Consider next a similar indicator function \( H_i : D \rightarrow \{0, 1\} \) for H-creativity such that \( H_i(x) = 1 \) indicates that an artefact \( x \in D \) is H-creative for individual \( i \). Here we encounter a problem that has not been adequately addressed regarding H-creativity—with respect to what context is \( H \) computed? Is it truly historical, meaning that \( H_i(x) = 1 \) indicates that individual \( i \) has created \( x \) for the first time in any context? Or, is it merely universal, meaning that \( H_i(x) = 1 \) indicates that, amongst the set \( I \) of all agents, \( i \) is credited with the “invention” of \( x \)? Or, is it limited further, such that \( H_i(x) = 1 \) indicates that, amongst some set \( C \subseteq I \), \( i \) is the creator of \( x \)?

For the moment, let us assume that \( H_i(x) = 1 \) indicates that, amongst some set \( C \subseteq I \), \( i \) is the first to have created \( x \). At one extreme, when \( C = \{i\} \),

\[
\forall(x), P_i(x) = H_i(x)
\]

However, as the size of \( C \) increases, how can we characterize the relationship between \( P \) and \( H \), or between P-creativity and H-creativity?

P-creativity represents creativity in the limited societal context of a single individual. H-creativity has been used to represent creativity in a more global societal context. Neither of these designations properly facilitate the characterization of creative behavior as it occurs simultaneously in the context of several nested and overlapping societal contexts (consider, for example, some of Lee Sedol’s group memberships, shown in Figure 3). In the context of a singleton group consisting of a single creator, all creative acts are both P-creative and H-creative [i.e., \( p(P(x) = H_i(x)) = 1.0 \)]; however, as additional creators join the group, the likelihood of any creative act by any individual being H-creative decreases (see Figure 4).²

Note that this decline in an agent’s ability to make “meaningful” creative contributions to the group may follow many profiles, as shown in the figure. The actual shape of the decline profile is both group- and individual-specific and will be affected by complex group dynamics, including membership demographics, group history, the domain in which the group is creating, group sociability, cohesion, and cooperation, to name a few. Exploring the relationship between these group characteristics and the shape of these contribution profiles suggests interesting research questions, but we commend those to future work.

Here, we focus instead on the fact that no matter the shape, the curves will always be monotonically decreasing with group size and note that here, too, there is a complex interplay involving group dynamics—for example, on the one hand, the larger the group, the less likely it is that a

---

¹It may be interesting to alternatively consider this to be a real-valued function \( G_i : D \rightarrow [0, 1] \), but for now we will consider the Boolean case, which may be more consistent with classical treatments of P- and H-creativity.

²This does not account for the sometimes confusing use of H-creativity in a temporal sense, as a measure of truly historic (and thus group-agnostic) creativity.
particular individual agent may make an important contribution; on the other hand, however, the larger the group, the more likely it is that significant creative advances are made by some agent, building on earlier the successes of group members. Again, we leave these kinds of research questions for the future.

In connection with the inverse relationship between likelihood of individual contribution and group size, we posit a satisficing threshold $\theta$ on $p(P_i(x) = H_i(x))$, specific to an individual, above which that individual is satisfied with their likelihood of contribution (also shown in Figure 4). Presumably, any group member gains some benefit as a member of the group—social connection, cooperation, learning, encouragement, challenge, critique, camaraderie; $\theta$ represents a cost that the member is willing to pay for these group benefits, and the intersection of the individuals contribution curve with their satisficing threshold, indicates a break-even point for group membership. We re-emphasize that both the threshold and contribution curve are group-relative and individual-specific. That is, they only have meaning relative to a group, and each group member has a unique valuation for group membership. Figure 5 shows one way to characterize creator type based on the interplay between an individual’s satisficing threshold for meaningful contribution and the shape of their likelihood curve for such contribution.

This idea of creativity relative to a specific community or group is related to ideas that go back at least to Csikszentmihalyi’s systems view of creativity, in particular his notion of field (1988). While other authors have since worked on operationalizing this social model in various ways, such work has focused on the behavior of the society (Sosa and Gero 2005) and on the society’s (dynamic) conceptualization of the domain (meaning, broadly, how it understands what is in the domain, what is valuable in the domain and how to explore the domain) (Linkola and Kantosalo 2019).

By contrast, what is wanted here is some way to talk about a society’s beliefs about itself. This is actually somewhat more related to Jennings’s social structure for creative agents, in which he employs a theory-of-mind and agent affinities; but, again, his aim is to explain creative behavior (2010).

We take an approach reminiscent of Jennings’s—given that we want to measure creativity relative to a community, it makes sense that the members of that community are involved in that measurement. In doing so, we offer a notion intermediate to P- and H-creativity that explores their reciprocal relationship, which we call group-level creativity or G-creativity.

G-Creativity

For a community of individuals, $C$, domain $D$, $i,j \in C$ and $x \in D$, define a family of indicator functions:

$G_{ij} : D \rightarrow \{0, 1\}$

that map artefacts from a domain $D$ to the Boolean set, indicating agent $i$’s belief in agent $j$’s creativity in producing artifact $x$. Note this judgement purposely does not differentiate a personal vs. group creativity—it is simply a belief indicator, parameterized both by critic and creator.

Given this family of indicator functions, and following Jennings’ example of generalizing beliefs, we can now compute a significant number of group-relative creativity measures that are more nuanced treatments of distinct ideas, which are conflated in the traditional conception of P- and H-creativity. This functional decline could take many forms (depending on $i$, $x$, $C$) but is always monotonically decreasing. The threshold $\theta$ represents a valuation of group membership.

Figure 4: The relationship between an individual’s likelihood of meaningful contribution and the size of the community of which the individual is a member. In a statistical sense, the larger the community, the less likely any single contribution for any single individual is considered H-creative. This functional decline could take many forms (depending on $i$, $x$, $C$) but is always monotonically decreasing. The threshold $\theta$ represents a valuation of group membership.

Figure 5: The satisficing threshold, $\theta$, and the shape of the likelihood curve for H-creativity on a scale from convex (e.g., the grey curve in Figure 4) to concave (e.g., the red curve) for a particular individual in a community characterize their group-mindedness and competitiveness in the community.
H-creativity. These new measures allow for both individual and collective creativity and beliefs about both types of creativity by both the individual and the group. We can marginalize any of the variables to produce these various viewpoints on creativity. For example, \( G_{ij}(\cdot) = \frac{1}{|D|} \sum_{x} G_{ij}(x) \) represents agent \( i \)'s belief about the creativity of agent \( j \), independent of artifact.\(^4\)

Perhaps the most salient viewpoints for the current discussion are an individual’s belief about their own (group-relative) creativity:

\[
G_{ii}(\cdot) = \frac{1}{|D|} \sum_{x} G_{ii}(x)
\]

and the community’s belief about an individual’s (group-relative) creativity:

\[
G_{jj}(\cdot) = \frac{1}{|C||D|} \sum_{i} \sum_{x} G_{ij}(x)
\]

The former is the viewpoint affecting Lee Sedol’s decision to retire from competitive Go—his belief about his own creative potential was adversely affected by his encounter with AlphaGo. This disruptive event introduced a precipitous drop in his estimation of his own creativity,\(^5\) driving it prematurely below his satisficing threshold. The latter viewpoint, that of the community’s estimation of Sedol’s creativity, was not so adversely affected; indeed, according to some commentators, Lee Sedol may in some way now be viewed as more creative, given his remarkable victory in game 4.

This reconceptualization of how we characterize creativity allows for a nuanced treatment of many issues, most particularly those raised here arising from the concern about the effect of dominant CC systems. The refined concept of P-creativity provides a socially relative formalization that spans the spectrum naturally demarcated by the classical concepts of P- and H-creativity. For a catalogue of concepts resulting from this formalization, see the Appendix.

**Discussion**

This more nuanced view of characterizing creativity, with a social lens, suggests many avenues for discussion, of which we mention a few that we find compelling.

First, because \( G \) is parameterized by community and individual, it can provide a rich representation of the concept of creativity. In particular, each such parameterization provides a different viewpoint. And, therefore, any characterization/evaluation by any entity (individual or group) of creativity should take into account these multiple viewpoints—the assessment of creativity is an agglomeration of multiple individual and group beliefs regarding the artifact, act, individual or group in question. In the context given here, we have so far mostly maintained a community-centric perspective—considering creativity from a single, fixed group membership; however, it is equally plausible to consider things

\(^4\)Operationally, the denominator of the normalization term will likely be approximated as \(|X|\), where \(X \subset D\).

\(^5\)Actually, based on one of the quotes above, his estimation of human creativity in general, represented by the viewpoint \( G^\text{human}_{ii}(\cdot) \), may have been adversely affected as well.

from an individual-centric one, recognizing that an individual may be a member of multiple communities. This would necessarily introduce additional generalized viewpoints of creativity, for example, marginalizing over an individual’s group memberships. In other words, all creativity has value in some community and failure to consider creativity from different viewpoints and across different societies results in a failure to effectively assess (an individual’s) creativity.

Second, in the context of a community of creators, it is not clear what group membership entails. What is required for an individual to feel like they are a member of the group, and what is required for the group to accept them as such? While a satisfactory answer to this question is beyond the scope of this treatment, for the purpose of discussion, we suggest that the answer is likely to include things like empathy/understanding, communication, cooperation, mutual admiration/inspiration. Given this, a natural question is whether a CC system can ever meet such a standard?\(^6\)\(^7\) Can a CC-system share common experiences with other members of a group? Or inspire or be inspired by other members of a group? Note, that while in the general case, such groups may consist entirely of artificial agents, we are here most interested in groups with human members. So, can a CC system inspire human creativity the way human creativity inspires human creativity? Does AlphaGo inspire? Or just discourage? Are those two versions of the same thing? Humans have an intrinsic (emotional?) connection with each other by virtue of sharing a common species, something they do not naturally have with computational systems. If CC systems are to be considered members of a group (that includes humans), is this kind of connection necessary, and is it something we really want to encourage? Also, assuming it is possible for CC systems to be members of a group, is the introduction of a computational agent into a (human) community somehow fundamentally different than introducing another human agent? If so, why? Why is Lee Sedol’s reaction to his experience with AlphaGo so different than his experience with losing to Lee Chang-ho? Why doesn’t this make him feel like he feels about AlphaGo?

Third, as CC systems continue to advance, they will begin competing for people’s jobs, especially in the realms of content creation; we have already seen the development of systems for producing music (Carré, Pachet, and Ghezdi 2017) and soundtracks (Brown 2012), video game assets (Hello Games 2016), logos (Sage et al. 2018) and slogan generation (Gatti et al. 2015), and news articles (Montal and Reich 2017), to name a few. As computational resources become cheaper and these systems become more advanced, it is likely that employment opportunities for human creatives will be negatively impacted. To the extent that we, as a field, care about the impact our work may have on society and the resulting attitudes society may have about our

\(^6\)This is currently a common critique of CC systems purporting to work in artistic domains—extant systems have no sense of community.

\(^7\)Interestingly, after AlphaGo defeated Lee Sedol, the Korean Baduk Association awarded it an honorary ranking of 9-dan, equal to that of Sedol, for its “sincere efforts” to master Go’s Taoist foundations and reach a level “close to the territory of divinity”.

Proceedings of the 11th International Conference on Computational Creativity (ICCC’20)
ISBN: 978-989-54160-2-8
work, it is important to consider these “mundane” creative impacts as well as the grander ones illustrated by cases such as AlphaGo. Perhaps as a field we should focus on domains in which supplanting people would be welcome (dangerous tasks or curing cancer); or on domains where finding qualified/interested human applicants is difficult (programming, teaching); or on nascent domains for which competition is low (virtual reality)? Certainly as a community we must accept the challenge to continuously and carefully articulate why CC is justifiable as a field of research—i.e., how does CC benefit humanity enough to justify its negative impacts?

Finally, perhaps the right approach is a focus on co-creative systems that complement rather than compete with human abilities or on systems that teach creativity. Imagine a system that could teach what it knows about Go. Perhaps one way to ameliorate some of the potential negative impact of a CC system is the requirement that the system be able to explain its creativity (Bodily and Ventura 2018). Is it possible that the assumption by CC systems of creative responsibilities that are primarily exploratory in nature could facilitate advances in transformational creativity by humanity, as humans move towards new forms of creativity that avoid competing with CC systems on unfavorable terms? Could such an evolutionary effect result in a change in the rate at which transformational creativity occurs? Could CC systems become the Iron Giants on whose shoulders the next Newtons stand?

**Conclusion**

We have argued that

(a) CC systems that supercede human-level creativity are becoming a reality, and it is urgent that the community begin thinking about the implications of this,

(b) (almost) all creative domains include some natural level of competition/comparison and therefore any creators in those domains are susceptible to being affected by dominant CC systems operating in that domain, and

(c) G-creativity offers a reconceptualization of the notions of P- and H-creativity that provides us a more fine-grained set of tools with which to wrestle with points (a) and (b).

We’ve demonstrated the utility of these tools in raising questions and making suggestions about the affects that dominant CC systems may have on their domains of expertise and on their human peers.

**References**


Conference on Computer Vision and Pattern Recognition, 5879–5888.


Appendix: Formalized concepts derived from G-creativity

Group-relative concepts of creativity that can be represented with a conceptualization of G-creativity. These concepts can be organized in various ways, and here we do so according to subject/object and individual/group. Given a community $C$, a domain $D$, agents $i, j \in C$ and artifact $x \in D$, we have the following:

A. Individual beliefs about individuals:

1. $G_{ij}(x)$ is agent $i$'s belief about agent $j$'s creativity when producing $x$.
2. $G_{ii}(x)$ is agent $i$'s belief about their own creativity when producing $x$.
3. $G_{ii}(\cdot) = \frac{1}{|D|} \sum_{x} G_{ij}(x)$ is agent $i$'s belief about agent $j$'s creativity in general (independent of a particular artifact).
4. $G_{ii}(\cdot) = \frac{1}{|D|} \sum_{x} G_{ii}(x)$ is agent $i$'s belief about their own creativity in general (independent of a particular artifact).
5. $\arg\max_{x} G_{ij}(\cdot)$ identifies the artifact $x$ that agent $i$ believes is agent $j$'s most creative work.$^8$
6. $\arg\max_{x} G_{ii}(\cdot)$ identifies the artifact $x$ that agent $i$ believes is its own most creative work.
7. $\arg\max_{x} G_{ij}(x)$ identifies the agent $i$ with the highest opinion of agent $j$'s creativity producing $x$; agent $j$'s advocate/champion for $x$.
8. $\arg\max_{x} G_{ii}(x)$ identifies the agent $i$ with the highest opinion of their own creativity producing $x$; the agent most certain they invented $x$.

B. Individual beliefs about the group:

9. $\arg\max_{x} G_{ij}(\cdot)$ identifies the agent $i$ with the highest opinion of agent $j$'s creativity in general (independent of a particular artifact); agent $j$'s overall advocate/champion/admirer.
10. $\arg\max_{x} G_{ii}(\cdot)$ identifies the agent $i$ with the highest opinion of its own creativity in general (independent of a particular artifact); the most self-confident member of the group.
11. $\arg\max_{x} G_{ij}(x)$ identifies the agent $j$ of whom agent $i$ has the highest opinion regarding their creativity producing $x$; the agent that agent $i$ believes invented $x$.
12. $\arg\max_{x} G_{ij}(\cdot)$ identifies the agent $j$ whom agent $i$ considers to have the highest creativity in general (independent of a particular artifact); the agent that agent $i$ most admires/reveres?

C. Group belief about the group:

13. $G_{ij}(x) = \frac{1}{|C|} \sum_{i} G_{ij}(x)$ is agent $i$'s belief about the community's creativity in producing $x$.
14. $G_{ij}(\cdot) = \frac{1}{|C||D|} \sum_{i} \sum_{x} G_{ij}(x)$ is agent $i$'s belief about the community's creativity, independent of a particular artifact.
15. $\arg\max_{x} G_{ij}(\cdot)$ identifies the artifact $x$ that agent $i$ believes is the community's most creative work.
16. $\arg\max_{x} G_{ij}(x)$ identifies the agent $j$ with the highest opinion of the community's creativity producing $x$; perhaps the community advocate/champion/promoter for $x$.
17. $\arg\max_{x} G_{ij}(\cdot)$ identifies the agent $i$ with the highest opinion of the community's general creativity; the community advocate/champion/promoter.

D. Group belief about the group:

18. $G_{ij}(x) = \frac{1}{|C|} \sum_{i} G_{ij}(x)$ is the community's belief about agent $j$'s creativity when producing $x$.
19. $G_{ij}(\cdot) = \frac{1}{|C||D|} \sum_{i} \sum_{x} G_{ij}(x)$ is the community's belief about agent $j$'s general creativity, independent of a particular artifact.
20. $\arg\max_{x} G_{ij}(\cdot)$ identifies the artifact $x$ that the community believes is agent $j$'s most creative work.
21. $\arg\max_{x} G_{ij}(x)$ identifies the agent $j$ of whom the community has the highest opinion regarding their creativity producing $x$; that agent that the community believes invented $x$.
22. $\arg\max_{x} G_{ij}(\cdot)$ identifies the agent $j$ whom the community thinks has the highest general creativity, independent of a particular artifact; the community “champion”.
23. $G_{ij}(x) = \frac{1}{|C|} \sum_{i} \sum_{j} G_{ij}(x)$ is the community's belief about its collective creativity in producing $x$.
24. $G_{ij}(\cdot) = \frac{1}{|C||D|} \sum_{i} \sum_{j} \sum_{x} G_{ij}(x)$ is the community's belief about its collective general creativity, independent of a particular artifact.
25. $\arg\max_{x} G_{ij}(\cdot)$ identifies the artifact $x$ that the community believes is, collectively, its most creative work.
Engendering co-creative experiences through agent parametric control

Prashanth Thattai Ravikumar
Department of Communications and New Media
National University of Singapore, Singapore
prashanth.thattai@u.nus.edu

Devangini Patel
School for Integrative Sciences & Engineering (NGS)
National University of Singapore, Singapore
devangini27study@gmail.com

Abstract

One of the important challenges in developing computational musical partners is to impart experiences that are different from musical support. In this work, we use an existing rhythm improvisation system to explore how we can engender different co-creation experiences by varying the system’s co-creative behavior. In an exploratory study, the system’s co-creative behavior was varied by manipulating a single experimental control and used to engender co-creative experiences in rhythm duets. In one experimental condition, the system induced a sense of producing divergent material, influencing creative outcomes, and negotiating musical outcomes, three aspects that were consistent with group music creativity. These findings provide supportive evidence that the system’s parameter can be varied to systematically engender co-creative experiences distinct from creative support.

Introduction

The development of semi-autonomous systems for creative partnership is an important topic in the domain of computational co-creativity. Creative partner systems align well with co-creative systems in their autonomous or semi-autonomous decision-making abilities, and engendering engaging user experiences. These characteristics suitably place them within the spectrum of fully autonomous systems that produce creative artefacts on their own, and creativity support tools that support the user’s creative thought (Davis 2013). In this work, we focus on engendering experiences of creative musical partnerships with co-creative music systems.

Prior work on the topic has focused on studying computational music partnerships in particular musical setups and improvisational genres. This includes work on systems that use autonomy and impart a perception of agency (Bown 2018), have a musical personality and sometimes engender a sense of partnership (Albert 2013), behave unpredictably and engender a sense of equal co-improvisation (Lewis 2000), and provide support with engaging interaction behaviors (Brown, Gifford, and Voltz 2016). In spite of this progress, one of the challenges for computational musical partnerships is to engender experiences of creative collaboration different from the experience of being creatively supported by an agent. A comparative analysis of such differences among systems is difficult as they often play different musical genres and have specific performance settings that are difficult to experimentally control and recreate. As a first step towards addressing these challenges in the study of creative musical partnerships, we present a study setup focused on engendering and studying different experiences of collaboration with the same music co-creation system.

The central question about partnerships addressed in this work is: How can systematic variation of an agent’s rhythmic interaction behavior engender different experiences of co-creation? The co-creative system, MASSE (Ravikumar and Wyse 2019) is used in an experimentally designed musical environment to study co-creative experiences. MASSE provides a facility to control the system’s co-creative behavior through a single parameter which could directly impact a musician’s experiences of collaborating in rhythmic duets. The central hypothesis is that the differences, if any, in human experiences with the agent configurations could be analyzed to identify factors that distinguish a musician’s sense of creative collaboration different from a sense of co-creative support.

In the rest of the paper, we review prior work on collaboration with rhythm improvisation systems and methods for studying co-creative experiences in artistic co-creation systems. Then, we present an experimental study design to study human experiences of collaboration with MASSE (Ravikumar and Wyse 2019). We report the observations of collaboration experiences of 4 expert musicians when they co-improvised rhythmic duets with different configurations of the system. The musicians’ responses to probes regarding various aspects of their experience from the study are qualitatively analyzed and the emerging themes are discussed through Sawyer’s framework of group creativity (Sawyer 2006). The study and the analysis highlight themes that productively contribute to a musician’s experiences of collaboration different from creative support.

Related Work

Related work briefly reviews other rhythm improvisation for partnership studies. Through an evaluation of these systems, we realized that there has been limited empirical validation with participants to understand how this engendering could possibly happen. Thus we widen our review of existing work...
to cover and techniques for the comparative study of co-
creation experiences with systems in other artistic and lin-
guistic domains.

**Rhythm improvisation systems**

For the purposes of the paper, we restrict our analysis to three music co-creation systems that provide the facility to vary their co-creative behavior in rhythmic improvisation. We focus on rhythmic improvisation as it allows for restrictions relating to musical tempo, a fixed metrical structure, a few timbres, and events quantized to beats that constrain musi-
cality but do not substantially reduce the complexity of the interactions. Three rhythm improvisation systems that co-
move with humans in such environments are described along with musicians experiences of playing with them.

One such system is the Clap-along system developed to interactively negotiate musical outcomes in a rhythmic duet (Young and Bown 2010). The system receives an input rhythm, and produces variations of the rhythmic onsets that simultaneously increase similarity with a target rhythm, and the musician’s input. Musicians who engaged in negotiating with the system felt a sense of negotiating towards a target rhythm when they predictably repeated their rhythms. How-
ever, they found it difficult to introduce rhythmic changes as the system produced rhythmic variations that were too di-
versgent and difficult for them to follow.

In contrast to Clap-along, the Ambidrum system uses de-
termistic mappings to produce rhythms that maintain a balance between coherence and novelty (Gifford and Brown 2006). The system produces responses by transforming the rhythmic input based on a correlation function. In its trans-
formation, the system modifies three aspects of the input rhythm, namely, intensity, pitch, and duration. Depending on the target correlation value, the system produces rhythms that vary on a range of behaviors from imitating to com-
plementing the inputs. In real-time performance, the system provides a slider that can be used to directly control the complementary of its responses. An avenue for improv-
ing the system’s unpredictability was to change target coher-
ence values to introduce fluctuations in its behaviors. How-
ever, there did not seem to be a follow-up of this work that demonstrated this. A possible challenge with using deter-
ministic mappings is that the musicians may begin to antici-
pate the system’s behavior ahead of time, which may not be perceived as co-creative.

The musical system MASSE (music action selection with state evaluation) (Ravikumar and Wyse 2019) uses a combi-
nation of techniques from the two prior systems. Similar to Ambidrum, the system uses a target correlation value that it maintains during the interaction. In order to produce rhyth-
mic responses, the system generates rhythmic variations and selects one based on evaluation functions. However, the sys-
tem is different from both the previous systems in that makes adaptive decisions to guide its behavior based on an author-
specified goal.

MASSE is specified with a system goal to maintain a per-
ceptual level of rhythmic stability and togetherness. In real-
time performance, the system assesses the combined musi-
cal outputs in terms of deviations from the goal. In response to this assessment, the system produces rhythmic output that brings the stability and togetherness levels back to the ex-
pected state. Through a goal state that drives system be-
avior, MASSE provides a facility to manipulate co-creative behaviors using a single parameter. As a proof of concept, MASSE was used to demonstrate differences in co-creative behaviors with a synthetically constructed lead-input. The system is yet to be tested with human subjects.

Although different rhythmic improvisations have been de-
veloped with parameters that can be varied to change system behavior, there seems to be a limited evaluation of a system’s ability to impact co-creative experiences in experi-
mental control. We widen the scope of our analysis of sys-
tems beyond music to identify methods and experimental setups to evaluate co-creative systems in other artistic do-
main.

**Comparing co-creative experiences in artistic systems**

Experimental work from computational co-creative is re-
viewed that has engendered different collaborations through experiments with human and non-human actors, and be-
ne mixed-initiative conditions.

The experimental study with the Drawing apprentice sys-
tem is perhaps closest to the notion of engendering and studying different collaboration experiences that is the focus of this paper (Davis et al. 2016). Davis and his colleagues created an art-based co-creation system that collaboratively creates free-form drawings with a human, and studied hu-
mans sense of their engagement with the system in two condi-
tions - Wizard-of-Oz, and the agent condition. The re-
ports from different conditions were compared to observe differences in human’s collaboration experiences. From this study, three themes related to participatory sense-making emerged in the human-human collaboration and were used to analyze the collaboration in the human-agent condition. The themes pertained to making sense of contributions, the dynamics of the interaction, and emergent meaning in the interaction. The method used to evaluate the Drawing apprentice is directly relevant to the work here. However, dif-
f erences in collaborations were engendered through interac-
tions between a human co-creator and co-creative agent, an aspect that is peripheral to the focus of this work.

Another approach that has been used in artistic systems involves engendering differences in collaborations in sys-
tems conditions of mixed-initiative. In Sentient Sketchbook, users collaboratively design the elements of a game level with an agent that provides suggestions for this task (Yan-
накис, Liapis, and Alexopoulos 2014). The experiments with Sentient Sketchbook investigated the degree to which users take suggestions of the computational co-creator in the follower condition. The authors evaluated how did the com-
putational suggestions affect the perceived quality of the so-
lutions. Results indicate that the computer-generated sug-
gestions are not used often when the human has initiative, but they can result in major changes in the maps’ appear-
ance. As an example, the computer-generated output breaks the visual patterns and introduces more imbalance. The pa-
per by Oh et al. 2018 identifies themes pertaining to the per-
ception of art-based co-creation systems with human users. In a mixed-initiative structured co-creation task, the human and the agent co-create artistic drawings (Oh et al. 2018). In conditions when the system leads, the human co-creator felt that they were being forced to move in a certain direction by the AI.

In summary, authors of some artistic co-creative systems have varied task-initiatives to observe different collaboration experiences. However, prior work does not focus on directly studying the differences between musical support and musical collaborations - an aspect that is of interest in our work. For the purposes of addressing the central research question, we present a system and study configuration.

Agent improvisation system

The central question about co-creative experiences that is of interest in this work is: How can the systematic variation of an agent’s rhythmic interaction behavior engender different experiences of co-creation? Among the rhythm improvisation systems reviewed in the related work, MASSE (Ravikumar and Wyse 2019) was selected for. Compared to other rhythmic systems, MASSE was selected for exploring creative collaboration experiences due to its technical and design considerations. From a design standpoint, the system was developed based on guidelines that were designed for imparting a sense of co-creation in minimal improvisation settings (Ravikumar and Wyse 2019). From a technical and provided a facility to vary system parameters to affect co-creative behaviors.

<table>
<thead>
<tr>
<th>Agent configuration</th>
<th>State-score range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low state-score</td>
<td>[0.0 - 0.2]</td>
</tr>
<tr>
<td>Medium state-score</td>
<td>[0.2 - 0.4]</td>
</tr>
<tr>
<td>High state-score</td>
<td>[0.6 - 1]</td>
</tr>
</tbody>
</table>

Table 1: Agent configurations and state-scores

At the onset of the interaction, the authors specify the goal that drives the system’s co-creative behavior. The system’s goal specifies the expected target value of rhythmic stability (of the agent’s rhythms) and togetherness (with the musician) that system maintains in response to the musician’s input (Ravikumar and Wyse 2019). Based on pilot tests, the authors found three configurations that produced noticeable differences in the system’s behavior varying from creative support and opposition. Table 1 refers to the internal numeric representation of goals in MASSE.

Characterizing experiences of musical co-creativity

In order to analyze musicians experiences of interaction with the system, Sawyer’s model of group creativity is selected (Sawyer 2006). Sawyer’s model describes several interaction-level characteristics important to group creativity practices and was selected as an appropriate level of analysis for studying co-creation experiences. Notably, Sawyer’s work on small group interactions has studied the characteristics that contribute to a sense of group musical creativity (Sawyer 2006). Improvising groups often talk about unpredictability in determining future outcomes, symbolic interaction through musical and extra-musical communication, and meta-pragmatics as negotiating subjective understandings of the musical emergent through interaction as an integral part of engaging in group creative interactions. In this work, the performances between the agent and the musician are analyzed through creative process in group interactions in music improvisation.

Among the factors that are identified in Sawyer’s work, we focus on three aspects of musical interaction that are important to musicians experiences of free-improvisation. Prior work that has studied the musicians attitudes and beliefs about improvisation identifies a sense of interaction through sound, sense of alternating musical roles, and negotiating musical characteristics as more important to a sense of co-creation over other aspects such as social interaction or musical skill (Ravikumar, McGee, and Wyse 2018). In the rest of the work, we focus on these aspects of musical interaction for analyzing and discussing musicians co-creative experiences.

Experimental design for studying co-creative experiences

Experimental Setup

Several constraints were added to ensure that the performances generated in the experimental setup were comparable in their musical aspects. These constraints were enforced in order to minimize the differences in the user’s experiences between the system conditions, and enable meaningful comparisons across conditions. The performances were constrained in terms of the structure of the musical piece, style and organization of the musical content, and the range of musical elements available for improvisation. In addition to these, musicians were also informed about about the limitations of the co-creation system as the expectations of the partner may influence the experience. The different constraints are described below.

Length and structure of musical pieces The musical pieces were limited to a period of 1.5 to 2 minutes. In each musical piece, musicians start a rhythm pattern, initiate two transitions to other rhythms (excluding the starting and the ending), and conclude the piece. The performances were focused around transitions as these are focal points of decision making during collective free improvisation (Canonne and Garnier 2012). During these moments, musicians are highly likely to expect co-creative interaction such as negotiating musical material, and influencing the behavior of other performers.

Maintain musical style While constructing each piece, musicians were instructed to play at a medium level of density (neither too high nor too low), maintaining a similar feel in the musical groove, and performing similar kinds of rhythm patterns that are similar in their musical feel. This was enforced to minimize the differences in musicians actions between performance trials and to improve consistency in actions across different musicians.
Restricted timbres and dynamics  The setup also involved restrictions on the range of musical elements available. Musicians performed rhythmic sequences that were constructed from eighth notes, quarter notes, and half-notes. The musician and the agent were each assigned one timbre with which they constructed the rhythms. A high timbre was assigned to the musician to subtly indicate that they were leading the performance, and the base timbre was assigned to the agent. Musicians were informed that there were no changes in the volume of the hits. This constraint was added to reduce the impact of the creative selection of the sounds and musical phrasing that may impact the experiences of the musician.

System restrictions  In addition to the musical restrictions, musicians were informed about the restrictions of the rhythm improvisation system with which they co-improvised duets. Prior to their interaction, musicians were informed that the agent listens to 2 bars of their playing, and responds with 2-bar rhythm patterns synchronized with the tempo. In all the settings, musicians were instructed that they would initiate the rhythmic material, and decide when to shift rhythms to which the agent responds. This was enforced to keep musicians focused on how the agent responds to their rhythms, and focus less on the agent’s ability to create musical material on its own. Finally, musicians were asked to evaluate interaction in the middle portion of the piece leaving the beginning and the ending. The system did not make decisions about the starting, or ending of the musical piece. Subsequently, musicians were not asked to evaluate the system’s behavior during these portions of the piece. With the above-mentioned restrictions, musicians performed rhythmic duets with the co-creative agent and reported their experiences.

Study setup

The study involved four musicians who played rhythmic duets with two configurations of the agent (high state-score, and low state-score). These two conditions were selected as we expected the agent’s behavior to be most divergent in terms of their co-creative actions, and likely to produce differences in co-creative experiences. The details of the study conducted are presented.

Participants  Four musicians took part in the study. Each musician had more than 10 years of experience of playing the percussion or other melodic instruments (e.g., saxophone, piano) and more than 4 years of experience improvising with other musicians.

Materials  The materials used in the study included a Korg controller for playing rhythm patterns, instructions for generating rhythmic duets (refer to Experimental Setup), and two versions of the co-creation system.

Musicians used a Korg controller to generate musical patterns and play with the system. The musician pressed the designated buttons on the Korg controller to trigger musical sounds. These sounds were played back along with the system’s response through a speaker. In this experiment, the musician was assigned the high timbre and the system was assigned the base timbre.

Two configurations of the system were used as conditions for the study. The system was configured to the low state-score and high state-score respectively as specified in Table 1. Musicians were exposed to the different conditions in a counter-balanced order.

Protocol  At the beginning of the session, musicians went through the training procedure in which they used the interface to play a two-minute music performance with the metronome. Each session lasted for 45 minutes to 60 minutes.

In the musical task, musicians played short rhythmic duets for 1.5 to 2 minutes. Musicians were instructed about the constraints of working with the system before the performance. Musicians began their performance by listening to the metronome bell and initiated rhythm patterns to move the improvisation forward. The system used a selected strategy to respond to the cues initiated by the musician. They were encouraged to initiate at least 2 transitions in the rhythm patterns during the performance.

Musicians played duets with both versions of the system that are presented in counterbalanced order. After each duet with the system, they answered questions about their interaction with the system through a semi-structured interview.

Data gathering  A semi-structured interview technique was used to gather musicians experiences of performing with the agent. In the interview, musicians described their overall sense of interaction with the system and answered several probe questions about negotiating the structure of the piece, musical characteristics of the performance, their sense of leading the interaction and developing rhythms together with the system. The interview guide used for these interviews is available in the Appendix.

Findings

The responses gathered from the musicians were qualitatively analyzed through thematic analysis (Attride-Stirling 2001). For the purposes of contrasting the differences in experiences, the themes that emerged in the different study conditions are organized along three main categories - sense of interaction, negotiating musical roles, and negotiating changes in the music (Ravikumar, McGee, and Wyse 2018). The main themes that emerged from the analysis of the different system conditions are explained and pictorially represented in Figure 1.

Producing divergent responses versus direct responses  Musicians reported that system produced a variety of responses in the high state-score condition compared to the low state-score condition.

In the high state-score condition, three musicians reported that the system produced divergent behavior as it responded to them. This was observed through their comments about the range of behaviors exhibited by the system. P2’s reports indicate that they engaged in several interaction behaviors other than mimicry. An illustrative quote from P2 was, “I did not feel like the system was trying to respond to me as
directly. Not that it would not respond to me as directly. It felt like, ratios. You know my four ideas of response. The ratios were different. The system was a lot more willing to ignore or contrast. Before this, it felt like it was trying to complement. The first one had a lot more contrast than this one. Even sometimes ignore”. P4 felt that the system produced divergent behavior through musical responses that were variations of their input. P4 said, “The system started making more interesting patterns. The system was feeding me the information I played before but musically more interesting”. These reports suggest that the system produced responses that related to musicians’ input but also diverged from it.

In contrast, musicians felt that the system was mostly mirroring them in the low state-score condition. This is illustrated through quotes about quality of systems responses to input, and its diversity of responses. P1 described that the system was adapting to the musical motifs that they played. P1 said, “I did not feel like it was playing many offbeat notes. Maybe it was because of what I was playing. I think it captured whatever my input was and supported that very well”. P2 reported the system’s behavior was less divergent in its response. P2 said, “This version of performance it seemed to hang on to ideas more strongly. It developed more closely with respect to the smaller language.” The second comment suggests that the system was not as divergent in its response, which made them feel that it was playing rhythms closer to what musicians were playing.

**Mutual influence versus individual control** Musicians reported differences in their sense of being influenced by the system’s playing in different conditions. Musicians felt a sense of being creatively influenced by the system’s actions in the high state-score condition, whereas they expected the system to adapt and follow their actions in the low state-score condition.

In the high state-score condition, three of the four musicians identified moments in the performance when they allowed the system to influence their musical decisions during the performance. During the moments when they were changing rhythms, P3 felt inspired to use the bass drum as material for the next section. P3 commented that “the system was not just bouncing off what they play but also influencing my decisions as a performer”. P4 also recognized a sense of mutual influence when they felt like starting a new section based on the systems’ response. P4 said, “The system played more interesting patterns that I listened to and wanted to start a new section in the piece, but did not do that as it would go beyond the constraints of the piece”. The above quotes illustrate the instances in the performance when musicians were listening to the system and being creatively influenced by its material.

In the low state-score condition, two musicians felt that the system creatively supported what they wanted to do in the piece. An illustrative comment from P3 on the system’s supportive behavior was, “It has been good as it is flexible in its support. It is like eh. Everything that I was after playing. Everything it did made sense. It supported what I was playing but also did not get in the way which I think is perfect for a duet.” Other themes that emerged in this condition indicate that system was a compliant follower in the interaction. A specific quote from P2 that illustrates this is, “It was like, I stepped on it and did not kind of play its way out. Instead, it responded and improviser does not need to do it. You can end up in a situation where you respond to plough through. If you step in it, it said, oh that happened, I’m going to do this. Which is very responsive but it does not have a lot of chutzpah”.

**Changing together versus following transitions** The third theme that emerged from the analysis highlights the differences in musicians’ sense of performing transitions to-
gether with the agent in different conditions. In the high state-score condition, musician expressed that the system’s actions induced a sense of taking part in the change whereas in the low state-score condition, they felt that the system followed their lead during a change. Illustrative quotes from musicians are used to support this observation.

While making changes, three out of the four musicians felt that the system arrived together in the transition in the high state-score condition. P1 felt that the system took part in the change and induced a sense of negotiating the transition. The specific quote was, “System prompted a change, I’d change along with it”. Yes, whenever I had more eight notes, the system would also play eighth notes, when the system started to decrease, I would decrease, and then it would decrease, but also, there was a disruptive element that prompted the change”. Other comments from P2 and P3 suggest that there was more than one moment when the system completed the transition together with them. A quote from P2 is used to illustrate this, “I really felt like we arrived together, which was really interesting and it did kind of the same thing again”. These are some quotes from the musicians that describe their experiences of coordinating transitions with the system.

In contrast, musicians felt a perceivable latency in the system’s response during the transitions in the low state-score condition. Two quotes from P2 and P3 are used to highlight this. When asked about the system’s response during changes, P2 reported that they sensed a perceivable latency in the system’s response. They said, “I was thinking of the lag bar. I do not know how quickly. I give it some musical input in the performance. It processes it in some way. It plays, supports what I’m playing. Its basically the lag between what I give it and what it responds that one or two bars, even what you play is getting resonated. It still works to support but I can hear the echo”. Another comment by P3 suggests that the system induced a sense that it was catching up to the musical changes initiated by them. P3’s specific quote was, “Maybe there is still that slight lag. Obviously, when trying to change the rhythm too quickly it might, its always trying to catch up, support. Like you said, it might be bar or two behind. That is only obvious when you are trying to change your rhythm a bit quickly”. These are some comments from musicians that support the observation that the system produced responses that were slightly behind the changes played by them.

In summary, the analysis of the two system conditions indicates three main differences that contributed to musicians’ sense of co-creation in the different conditions. Musicians reported a sense of co-creation when the system produced musically related yet divergent material, influenced their material during the performance, and induced sense of taking coordinating transitions together.

**Discussion**

The observations from the collaboration condition are discussed in the wider context of group creativity and consistent comparisons are drawn. Three aspects of Sawyer’s framework are used for this discussion, namely, emergent symbolic interaction, unpredictability in musical direction, and negotiation inter-subjective understanding (Sawyer 2006).

**Emergent symbolic meaning**

The first aspect that Sawyer notes is that of *symbolic aspects of communication in the interaction*. The system for rhythmic improvisation generated rhythmic responses by transforming parts of the input played by the musician. Musicians who performed with the system reported that the system engaged in complementary, contrast behaviors or even sometimes ignored the musician. One of the interesting observations of the interaction was that even though the system’s strategy for response remained the same, musicians ascribed different behaviors to system based on their own actions. This is interesting as it raises the possibility that the meaning of musical interaction behaviors is situated in the context of the interaction rather than encoded in the system or described in the musician’s actions. The meaning of symbols, that musicians described, were emergent from the dynamics of the interaction.

**Unpredictability in the musical material**

The second aspect of group creativity that is highlighted by Sawyer is the notion of unpredictability. In group creative performances, musicians have several choices to make in terms of the musical trajectory of the performance, and often do not know beforehand which trajectory will be chosen. In the high state-score condition, musicians reported that listening to the system influenced them to change musical trajectories that they could not have predicted before.

A possible reason for the emergence of this unpredictability is the system’s contrasting behavior in the high condition. In this condition, the system generates a rhythmic response that contrast the stability and togetherness of rhythms. In order to construct a response rhythm, the agent generates a set of musical choices that vary in the onset positions of the input rhythms. Rhythms that are selected to contrast stability and togetherness are more likely to have onsets that are in syncopated positions, unstable with respect to the meter, and do not overlap with the input. This difference may have introduced a sense of unpredictability in rhythmic development, at least in our experimental setup.

**Negotiating inter-subjective understanding**

The final aspect that Sawyer notes is that of *meta-pragmatics* that involves negotiation of inter-subjective understandings. In order to do this, musicians’ evaluate their successive contributions with respect to the emergent, and together make decisions about whether to include the contributions of others in the emergent. The agent set to the high state-score condition sometimes induced a sense of negotiation of the rhythmic change by taking part in deciding musical material (e.g., eighth notes, moments of silences).

During the moments of rhythmic change, the agent observes deviations in state-scores of the combined sequences from the expected target state-score. These deviations may be a result of changes in either or both stability and togetherness. In response, the agent generates and selects from a
new set of musical rhythms with contrasting levels of stability and togetherness. This may have induced a sense that the agent wanted to change the musical characteristics that were introduced for change. Through decisions that alter the stability and togetherness of moment-to-moment contributions, the system induces a sense of rhythmic negotiation during moments of change, at least in this experimental setup.

Conclusion

This research started with the question: How can a music system engender different experiences of co-creation in a musical performance. In order to study this, an existing music co-creation was configured in two conditions that were expected to be most divergent in terms of their co-creative actions. Musicians performed rhythmic duets with the system and reported their experiences. The analysis of the reports produced themes that differed in factors that were important to a sense of musical co-creation. A wider analysis of the work with regards to group creativity raises three key observations regarding the study of collaboration experiences and the design of musical systems that engender them. The findings from the studies inform the partner characteristics that are critical to musicians sense of co-creation different from creative accompaniment support (in the low state-score condition). In the high state-score condition, musicians sensed that the system produced divergent responses, creatively influenced their musical outcomes, and coordinated transitions together. These characteristics were different from the low state-score condition in which the system behaved predictably, and let the musician lead the interaction.

Furthermore, the co-creative aspects of performances in high state-score were consistent with the characteristics of group creativity. In the high state-score condition, the system generated co-creative behaviors that induced a sense of emergent meaning, unpredictability in rhythmic trajectories, and sense of negotiation during rhythmic changes. These observations are an initial validation that the combination of high state-score and rhythm generation procedure and is effective in engendering performances that exhibit aspects of group creativity, at least in our experimental context.

Finally, the conditions in which the system engendered different co-creation experiences correspond to two extremes of a parameter that was varied in the system design. The different versions of the system that engendered different co-creative experiences were generated by varying a single parameter, i.e., the target state-score range of the system. The findings from the study provide support to the claim that state-score computed from evaluation metrics may be varied to engender differences in co-creative experiences.

A potential direction for improving agent design could involve the development of agent initiative for generating musical material and studying its impact on musicians sense of creative partnership. Other avenues for improving the system evaluation involve using a greater number of participants for improving generalizability and exploring the middle ranges state-score to observe whether the different experiences of co-creation lie on a spectrum.

Semi-structured interview guide for co-creation questionnaire

1. What were the moments that worked for you in the performance?
   - What was happening in those moments that worked?
   - Were there other moments that worked?

2. Were there moments that did not work for your performance?
   - What were you expecting in those moments that did not work?
   - What has happening in those moments that did not work?
   - What could have made it work?
   - Were there other moments that did not work?

3. Probes:
   - Can you describe the different sections of the piece?
     - Did the different sections come together?
     - Were there things that did not work about them?
   - Can you describe the transitions between the sections?
     - Did the transitions work?
     - Were there things that did not work about transitions?
   - What worked about leading the system through the piece?
     - Were there things that did not work?
   - Did the different sections come together?
     - Were there things that did not work?
     - What were you expecting in those moments that did not work?
   - What worked about playing rhythms together with the system?
     - Were there things that did not work?
     - What were you expecting in those moments that did not work?

4. Based on your sense of the moments that worked, and did not work, how would you rate the overall sense of co-creation with them? (1 - rarely felt), sometimes, about half the time, more than half, 5 (always felt)

5. How important are each of the things you identified in the performance important to your sense of co-creation? (1 - not important at all, 2 - of little importance, 3 - of average importance, 4 - very important, 5 - absolutely essential)

6. Are there any other thoughts that you would like to express about the performance, your interaction with the system?
References
Modalities, Styles and Strategies: An Interaction Framework for Human–Computer Co-Creativity

Anna Kantosalo1, Prashanth Thattai Ravikumar2, Kazjon Grace3 and Tapio Takala1

1Department of Computer Science, Aalto University School of Science, P.O.Box 15400, FI-00076 AALTO, Finland
2Communications and New Media, National University of Singapore, 21, Lower Kent Ridge, Singapore 119077
3Design Lab, The University of Sydney School of Architecture, Design and Planning, 148 City Road, Darlington NSW 2008 Australia

Abstract

Human–computer co-creativity research is relatively new, and comparing how co-creative systems augment creativity is challenging even within the same creative domain. This paper proposes a framework to facilitate such comparisons, drawing on domain-agnostic concepts in interaction design research. It describes the different aspects of the interactions between one or more humans and their computationally creative collaborators at three levels: interaction modalities, interaction styles, and interaction strategies. Modalities are the channels by which information is exchanged, styles are the different behaviours and models that govern the system’s actions over those channels, and strategies are the goals and plans that shape those behaviours. The paper ends with an analysis of nine co-creative systems, demonstrating how the framework makes the comparison of very different systems possible.

Introduction

Human–computer co-creativity is a subfield of computational creativity dedicated to the study and design of interactive, creative computational collaborators. These computationally creative collaborators interact with humans through a user interface, and together they form a co-creative system. Co-creative systems are distinguished by their shared locus of creative initiative. They represent a middle ground between autonomous creative systems, which are intended as the sole shepherds of their own creativity, and creativity support systems, which instead facilitate the creativity of their users. In recent years computationally creative collaborators have been designed and implemented for many creative domains, including drawing (Davis et al., 2015), poetry (Kantosalo et al., 2015), and music (Ravikumar and Wyse, 2019).

This interstitial and emerging research field has a number of open questions including how to design (Kantosalo et al., 2014), evaluate (Kantosalo et al., 2015; Karimi et al., 2018b) and describe (Kantosalo and Toivonen, 2016; Davis et al., 2015; Yannakakis, Liapis, and Alexopoulos, 2014) co-creative systems. Put simply: co-creativity is a goal as broad as creativity itself, and made yet more complex by the many and varied ways humans and computers can interact. As the field matures it will be necessary to cohere around a set of domain-general ways to describe the capabilities of co-creative systems. One attempt at finding a unified framing for co-creative systems has been to look at how users and systems behave while creating (e.g. Kantosalo et al., 2015; Karimi et al., 2018b; Long, Jacob, and Magerko, 2019). This paper strengthens that tradition with a framework for describing interactions with co-creative systems.

Interaction design—a human-computer interaction (HCI) discipline for designing interactions with products and services—has been suggested as one way to frame the design (Kantosalo et al., 2014) and evaluation of co-creative systems (Bown, 2014) in a behaviour-centric way. We extend this suggestion, drawing several terms from the interaction design literature to construct a framework for describing the behaviour of computers and users within human–computer co-creative systems. The goal of the framework is to equip co-creativity researchers with a domain-agnostic vocabulary to discuss the capabilities and shortcomings of existing and proposed interfaces for co-creation.

Our framework draws from three traditional interaction design concepts: interaction modalities, styles, and strategies. Together they form a layered description of interactions between a human and a computationally creative collaborator, from the most elemental level describing different qualities of their interaction channel, right through to the higher-level reasoning processes that govern how the computer selects and achieves its objectives.

We begin with a brief discussion on the background of these key interaction concepts. We then move on to defining our three interaction layers, discussing how they have been used in HCI as well as how they apply to co-creative systems research. We then show how this framework facilitates the analysis of nine co-creative systems. These systems are selected not as a comprehensive review of the current state of the art in human–computer co-creativity, but as a demonstration of the breadth of our framework. We end with a discussion of some new areas of co-creative systems research that our framework suggests are as-yet underexplored.

Background

As recently as the mid 1970s, the notion that computer systems should be in any way designed to be easy for their non-expert users to operate was largely treated with derision (Carroll, 1997). HCI as a research field and interaction design as its associated professional design domain have arisen
in the four decades since. New kinds of design processes were needed for the digital era, as neither a focus on visual communication (as in graphic design) nor on physical consumer products (as in product/industrial design) were sufficient to tackle the design of digital products and services. Interaction design and HCI focus on the experiences evoked by digital products as well as their utility and aesthetics.

Interaction can have multiple purposes, such as communication of information, chat chat, refining common goals through discussion, planning a strategy, or giving commands (Barfield, 1993, pp. 208-209). It can also take many forms, including exchanging information with a system or direct manipulation of the properties of the system (Barfield, 1993, pp. 207-208). Research in the field of interaction design has sought to characterise the ways people perceive and interact with complex as well as creative (Shneiderman, 2007) tasks, providing literature which we draw on to construct our framework below.

Three terms emerge from interaction design literature for describing the interactions between humans and computers at different levels of abstraction. The first, modality, considers the physical properties of the interaction, focusing on information channels and nature of exchanged information. The second, interaction style, focuses on the design of the interface and how it supports different conceptual models and behaviours related to interaction. The final, interaction strategy, considers more abstract elements of interaction, such as how the different interaction styles evolve throughout interaction. These three concepts form the basis of our framework and we shortly discuss their use in interaction design literature below.

The HCI origins of interaction modalities

On a basic level, interactions between a human and a computer are limited by the affordances of the interaction participants, such as the input and output devices available to each party. Questions related to the selection of interaction modalities are studied especially by researchers interested in multi-modal interaction (Granström, House, and Karlsson, 2013, p.2). Different modalities can affect how additional or different information can be conveyed (e.g. Bernsen, 2002), as well as make interaction more natural and immersive (O’hara et al., 2013; Tham et al., 2018).

The notion of modality is a useful ground zero for describing co-creative systems. The key question here is what are the input and output channels of the human and computational collaborators? With the basic possibilities of a specific co-creative collaboration so established, it is then possible to examine their nature further.

The HCI origins of interaction styles

Interaction style is a somewhat loosely used term dealing with how communication happens between the system and its user. For example, it can be “conversational” or “direct manipulation” (Barfield, 1993, p.215). In graphical user interfaces interaction is based on “widgets” such as menus, forms, dialog boxes, or icon sets (Sutcliffe, 1988, pp.68-75). Other modalities, such as speech (natural language) and embodied motion (gestures) also make their own interaction styles. In this paper we adopt the definition by (Hix and Hartson, p. 57): “Interaction styles are a collection of interface objects and associated techniques from which an interaction designer can choose when designing the user interaction component of an interface. They provide a behavioural view of how the user communicates with the system.”

Similar discussions of the ways, both broad and narrow, how interactive systems can behave during co-creation are almost entirely absent from this community’s description of its systems. Where such descriptions do occur, such as in (Bray and Bown, 2016), they use bespoke terminology, making comparisons to other systems and other domains challenging. The field of computational creativity tends towards describing its systems in terms of their generative capacity and their evaluation metrics. Co-creativity researchers often add to that a description of what we would call their systems’ interaction modalities, especially as regards the communication between user and agent. What is missing from these descriptions is a generalisable way of discussing interaction style: how do the user and the agent collaborate? And what impact does that choice have on creativity?

The HCI origins of interaction strategies

Where interaction styles describe the way a co-creative system interacts with its user to perform creative tasks, interaction strategies govern why it chooses one available action over another. At its simplest this is expressed through evaluation metrics such as value, quality, novelty, surprise, diversity, and so on. These cornerstones of computational creativity have extra importance in co-creativity research because of the need for collaboration: ignoring the user and searching for optimality does not make a good creative partner. It is therefore beneficial, in many cases, to adapt the search towards what the user appears to be doing. This has the potential to be more complex than adding additional evaluation metrics, and may include personalising or prioritising the evaluation process, or even a meta-search for new metrics or generative procedures (Wiggins, 2006).

The term “strategies” has been used to describe how interactive intelligent systems (which co-creative systems can be considered a kind of) adapt. It has been applied to describe how conversational agents might take different approaches based on past user behaviour (Schuller et al., 2006), and similarly to govern what educational content to deliver based on estimates of student mastery in intelligent tutoring systems (Al-Nakhal and Naser, 2017).

In our framework we define the term “interaction strategy” to refer to the system’s evaluative metrics, its underlying goals, and the meta-reasoning process (Cox and Raja, 2007) by which it adjusts its metrics to best achieve those goals. Many systems do not implement all three parts of that definition, which we consider an underexplored and promising direction for future systems.

A Layered Framework for Interactions with Computational Collaborators

We have adapted the above concepts for human-computer co-creativity. As shown in Figure 1 they form a three layered
framework for describing interactions with a computationally creative colleague. The first layer, *interaction modality*, describes the channels and media of the interactions. The second layer, *interaction style*, builds on the modality, focusing on conceptual interaction and behaviours. The final layer, *interaction strategy*, allows for designing and describing systems with more elaborate goals and interaction plans, than traditional productivity systems.

### Interaction Modalities

**Definition:** Interaction modality describes the medium of communication implemented through one or more sensory channels. There may be specifications related to each channel, such as restrictions or constraints on the flow of information through each channel.

Interaction modalities describe the fundamental elements of sending and receiving information between a human and a computer. We consider them as unique combinations of different input and output devices, information channels, and sensory modalities. While interaction can be unimodal, using only one modality, interactions are usually multimodal, combining different modalities (Bernsen, 2002). Multimodality is used to increase the naturalness of human–computer interaction, to provide different types of information (Granström, House, and Karlsson, 2013, p.1) and to reduce errors in productivity systems (Jaimes and Sebe, 2007), but it also plays a role in creativity support. For example the modalities draw, write, talk and gesture have all been found important for supporting collaborative design (Eris, Martelaro, and Badke-Schaub, 2014).

**Sensory modalities** relate interaction modalities to human senses, such as hearing, smell, sight, touch, taste (Jaimes and Sebe, 2007; Schuller et al., 2006), the sense of balance (Schuller et al., 2006) or more obscure senses like the senses of temperature, kinesthetics, or vibrations.

**Information channels** are often associated with sensory modalities: The sense of sight typically relates to the visual channel, sense of hearing to the auditory channel, etc. (Schuller et al., 2006). The same sensory channel can be used to carry different kinds of information, and some information is suitable for multiple channels. For example the visual channel can be used to carry images or linguistic information, but linguistic information could also be carried over an auditory channel. Information can also be mapped or transposed to different channels as part of various representations, such as in a musical stave or a choreographer’s notes. Channels may be simplex or duplex, and there is no requirement for the channels used in a co-creative system to be symmetrical between human and computer.

### Interaction Styles

**Definition:** Interaction styles describe how interaction is structured between the human and the computational collaborator. They describe the inter- and intra-part relationships which distinguish different periods of interaction from one another. Multiple styles can be manifested in a single system. Styles build on the symbolic representation of objects, linking them with different conceptual models of interaction and behaviours of the computational collaborators.

Interaction styles are design paradigms describing how users interact with a product. They can be used as tools for thinking how different conceptual models of interaction could be realised through different interaction modalities (Rogers, Sharp, and Preece, 2011, p.206). With co-creative systems, we consider that an interaction style emerges from the interplay of a conceptual model of interaction and the behaviours of a system. The interaction between the two is mediated by objects that provide a shared context for the system and the user.

**Objects** act as a conduit between interaction modality and the rest of the interaction style. They are representations of important shared concepts and interface control elements that can be manipulated or interacted with during co-creation. They are often multimodal and may include an interpretation of the meaning of the object in addition to its physical representations in a medium\(^1\). For example the creative artefact under construction may be represented as an

---

\(^1\)Medium is at times used as an aspect of modality describing the physical realisation of information at the human–computer interface (see Bernsen (2002)). Due to the possibilities of mixing the term medium with the artistic medium, we have chosen the word
object. Such objects often have a specific iconic representation like the notation used in sheet music, or the layout of a poem on a piece of paper. The full understanding of these visual objects requires knowledge of the representation style and possibly a translation to another modality, such as audio.

Behaviours describe the actions of the human and the computational collaborator during interaction. These actions are linked to specific objects on the interface. They can involve concrete manipulation of the objects, be initiated through an object, or they can relate objects to each other. For example a human could directly manipulate a word object in a co-creative poetry system by editing the word, then request validation of the rhyme from the computer by clicking a button object, and finally the computer could return a validation of the word object, with respect to the context provided by other word objects in the poem object.

Conceptual models structure the interaction between the human and the computer. Example behaviours of co-creative collaborators include shadowing, mirroring, coupling and negotiating interactions (Young and Bown, 2010), operation based, request based and ambient interaction (Bown and Brown, 2018) and additive and iterative interaction styles (Clark et al., 2018). General terms like highly encapsulated systems, direct manipulation and programmable interfaces can be used for describing different levels of control and visibility in human-computer co-creative systems (Bray, Bown, and Carey, 2017).

For the purpose of analysing interaction between system and the user, conceptual models are classified into contribution structures, and interaction models. In additive interactions, the human and the system combine their contributions to build a shared object, whereas in interactive interactions, the human and the computational system evaluate their contributions to successively refine a shared object (Clark et al., 2018). Interaction models are characterized as operation based, request based and ambient interaction models of interaction by Bown and Brown (2018). In operation-based interaction styles, human users directly manipulate a generative algorithm to produce outcomes. In systems that use a request-based interaction style, human users submit requests to a system that returns results. In ambient interaction systems, the system listens to the user’s actions and generates its responses through autonomous meta-creative processes that run in the background.

Interaction Strategies

Definition: A co-creative system’s interaction strategy governs what it values at any given moment, how those values change, and why. We consider an interaction strategy to consist of metrics (how a system values creative artefacts), goals (the motivational drives by which a system alters its metrics), and metareasoning (the way a system derives metrics from its goals).

The notion of an interaction strategy gives us a way to examine different goals of a computational collaborator, going beyond the discussion of its evaluation metrics. We intend the term to subsume both its criteria for determining what is good and its reason for adopting those criteria. We want to declare an open bias in constructing our framework this way: we want the discussion around co-creative systems to more seriously entertain adaptive evaluation metrics that adapt to user behaviour. Separating goals and metrics allows us to describe systems that adapt their generative capacity to the user and/or their creations. Such systems might, for example, switch between different metrics (or even generate whole new metrics) in order to encourage their human collaborators in one direction or another.

Metrics are a core component of research in both autonomous and collaborative creative systems (Jordanous, 2012). They give creative systems a way to discern valuable creative behaviours. Much of the formative discussion in the first ten years of ICCC revolved around what metrics were necessary and sufficient for evaluating creativity. These questions also apply to co-creative contexts, although with some nuance due to the presence of a human collaborator. We see metrics in co-creative systems as falling into three broad categories:

1. Metrics of value: evaluation of the quality of the resulting artefact, including appropriateness, utility, and aesthetics.
2. Metrics of novelty: evaluation of the divergence of resulting artefacts, including surprise and diversity.
3. Metrics of interaction: evaluation of the user or the way the user and the system are behaving together, rather than evaluations of the artefact directly.

Many co-creative systems have only value metrics, relying on stochastic processes and human curation to inject meaningful divergence into creative artefacts. Such systems can suggest novel artefacts to humans, but not reason or meta-reason about novelty themselves, as they have no way to detect it.

Goals—as a seminal topic in AI research—can play many roles in creative systems. Here we focus on strategic goals, defined as a desired state that the computer aims to bring about by varying its evaluation metrics. A system with fixed evaluation metrics has only implicit goals: they are hardcoded into its construction. By contrast, goals may be explicitly represented within an agent (and therefore potentially reasoned about). Goals can also be interactive: able to be manipulated by the user through some interface, giving control over the interaction strategy to the human.

Meta-reasoning in an interaction strategy refers to the process by which a co-creative system arrives at its current evaluation metrics given its current goals. Meta-reasoning describes how an agent reasons about (and affects) its own reasoning processes (Cox and Raja, 2007). In our framework, the system reasons about how to adapt its evaluation metrics to achieve its goals. Framing metareasoning as part of an interaction strategy adapts research in creative meta-search (Wiggins, 2006) to a co-creative context. In the co-creative case meta-search does not occur in isolation, because there are multiple creative agents (at least one human and one artificial) that are attempting creative search in parallel. This means that an agent may need to adapt its search space not only to the progress of its own search but to the
behaviour of its human collaborator(s) (see e.g. Kantosalo and Toivonen (2016)). There are several possible “levels” of metareasoning for co-creative systems:

- **No metareasoning** occurs without explicit goals.
- **Metric prioritisation** occurs when a system weights (or otherwise chooses between) its evaluation metrics given the current context.
- **Metric formulation** occurs when a system creates or modifies an evaluation function, like through meta-search.
- **Goal reasoning** occurs when a system uses higher-level goals or drives to adapt its goals, which in turn allow it to adapt its metrics by one of the above methods.

Those last two categories of metareasoning are speculative: we do not know of any systems that are best described in that way. They are not, however, unimaginable: goal reasoning could be governed by more abstract motivations like curiosity (Grace and Maher, 2016), autonomy (Jennings, 2010), goal-awareness (Linkola et al., 2017), or intent (Cook and Colton, 2011). Such discussions are a regular feature in autonomous creative systems, and this framework provides a place for them in co-creativity.

### Analysing Co-Creative Interactions

To examine how interaction modalities, styles and strategies are utilised by human-computer co-creative systems, we analysed nine systems from different application areas of human-computer co-creativity. Three of the systems, Masse (Ravikumar and Wyse, 2019), Shimon (Weinberg et al., 2009) and MiMi (François, Chew, and Thurmond, 2007) are all interactive improvisation systems focusing on different types of music. While Masse and MiMi synthesise music, Shimon robotically plays the instrument marimba. Three systems, the Poetry Machine (Kantosalo et al., 2015), LyriSys (Watanabe et al., 2017), and an unnamed story generation system that we hereafter refer to as “Clark et al.’s story system” (Clark et al., 2018) represent different interfaces in the domain of linguistic co-creativity. The first focuses on poetry, the second on lyrics, and the final on the collaborative writing of prose. The final three systems are drawn from other domains of human-computer co-creativity, including drawing (Creative Sketching Apprentice (Karimi et al., 2018a)), improvised dance (ViewpointsAI (Jacob et al., 2013)) and game level design (Sentient Sketchbook (Liapis, Yannakakis, and Togelius, 2013)). The systems and their domains are summarised in Table 1. To demonstrate the capacity of our framework to describe and compare co-creative systems, we examine the modalities, styles and strategies used by these tools below.

### Modalities in the Sample Systems

The modalities used by the different systems can be divided into modalities focused on conveying domain specific information and modalities conveying other information. For example, the MASSE system only conveys music specific, intra-musical information through a full-duplex auditory channel and a tactile input channel consisting of a

<table>
<thead>
<tr>
<th>System</th>
<th>Reported in</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>MASSE</td>
<td>Ravikumar and Wyse (2019)</td>
<td>Music</td>
</tr>
<tr>
<td>Shimon</td>
<td>Weinberg et al. (2009)</td>
<td>Music</td>
</tr>
<tr>
<td>MiMi</td>
<td>François, Chew, and Thurmond (2007)</td>
<td>Music</td>
</tr>
<tr>
<td>Unnamed Story Generation System</td>
<td>Clark et al. (2018)</td>
<td>Linguistic</td>
</tr>
<tr>
<td>LyriSys</td>
<td>Watanabe et al. (2017)</td>
<td>Linguistic</td>
</tr>
<tr>
<td>Sketching Apprentice</td>
<td>Karimi et al. (2018a)</td>
<td>Drawing</td>
</tr>
<tr>
<td>Viewpoints AI</td>
<td>Jacob et al. (2013)</td>
<td>Dance Game level design</td>
</tr>
<tr>
<td>Sentient Sketchbook</td>
<td>Liapis, Yannakakis, and Togelius (2013)</td>
<td>Game level design</td>
</tr>
</tbody>
</table>

Table 1: Analysed co-creative systems and their domains.
tems Poetry Machine and Lyrisys, as well as the Sketching Apprentice and Sentient Sketchbook.

**Interaction Styles in the Sample Systems**

There are several different conceptual models; ambient, request-based and operation-based interaction, among the examined systems. These connect to different objects and behaviours within the systems.

Ambient interaction is used by MASSE, Mimi, Shimon, Viewpoints AI and the Sketching Apprentice. The first two musical systems, MASSE and Mimi, use ambient interaction models with additive contributions that are simultaneously played with the input. Both systems listen to a predetermined number of musical bars before they start to play along with the musician. The rhythmic objects generated and listened by MASSE are 2-bars in length, while the audio input and visual music notation output of Mimi are arbitrary in length.

Shimon, Viewpoints AI, and the Sketching Apprentice, use an ambient interaction model to produce additive contributions in a turn-taking fashion. The objects used by the systems are domain specific: Shimon listens to musical objects, the Viewpoints AI gathers stylized human movements, and the Sketching Apprentice uses visual line drawings as interaction objects. Corresponding objects are used by the systems to listen for interaction events to decide when their turn begins. Viewpoints AI and the Sketching Apprentice are similar in their interaction behaviors and contribution types, producing interaction behaviours through transformation functions, while Shimon predominantly attempts to imitate the style of the musician. Then again, both Shimon and Viewpoints AI produce contributions that combine with the user input and are additive, while the Sketching Apprentice system generates suggestions to refine a creative artefact, showing an example of iterative contribution.

A request-based model of interaction is used by Poetry Machine, Clark et al.’s story system, and the Sentient Sketchbook. The users of these systems modify a shared artefact, and send requests to the system for feedback or additional materials. The two linguistic systems allow the user to directly manipulate object representations through a WYSWYG editor, whereas the Sentient Sketchbook provides a tile-based interface for the user to make design changes to the game world. All systems offer means for direct user manipulation. In the Poetry Machine system, the user directly makes changes to poem text and sends a request to the system to validate the rhyme, or request additional materials. In response, the system returns a validation of the poem, or materials suitable for a specific context. In the story generation system, the user directly manipulates the sentences that are automatically generated for every line and adopts them into the story. The Sentient Sketchbook system uses similar methods and also allows the users to directly manipulate the game-level, send requests, and adopt the suggestions generated by the system.

The final system, Lyrisys, is an example of a system that uses an operation-based model with direct manipulation of the interface objects. The Lyrisys system provides interface objects as parameters (e.g., syllable count, lyric and story) that the user manipulates to generate the lyric. The user changes parameters and pushes a button to generate the lyrics that are consistent with other context. In contrast to the previous systems, the user makes no changes in the content of the artefact but observes the behaviours of the system in response to parametric changes.

**Interaction Strategies in the Sample Systems**

The musical systems Shimon and Mimi both use a static interaction strategy to select their musical responses: they have implicit goals and no meta-reasoning. Mimi does provide an interface that allows the performer to evaluate the contributions of the system and change the preparatory musical material used by the computational collaborator, but its metrics remain unchanged by this configuration. Mimi has no metrics, being purely stochastic, while Shimon uses an interaction metric: the degree to which it is musically conforming with its collaborators. The MASSE system, by contrast, has an explicit goal to maintain a target level of stability and togetherness in the performance. It senses deviations from the goal state and sets targets for its, a kind of metric prioritisation. It has two metrics: stability (a value metric), and togetherness (an interaction metric).

The Poetry Machine and Clark et al.’s story system both have value metrics: rhyme, meter and alliteration for the former, and word-level conditional probability for the latter. Both can generate novel artefacts stochastically, like Mimi and Shimon, but do not have novelty metrics. Both also have implicit goals and no meta-reasoning. Lyrisys also has value metrics (meter, rhyme, and topicality) but differs in that the human can interactively select a goal topic.

Viewpoints AI has a variety of response modes that it switches between randomly: repeating the user, picking a novel gesture from its library, or transforming it (by reflecting a limb, switching limbs, or duplicating movements to other limbs). In various modes it applies a tree-based gesture similarity metric to either conform with (an interaction metric) or diverge from (a novelty metric) the user. Each mode is effectively a separate generative system, and its random switching between them means it has no explicit meta-reasoning. The Sketching Apprentice has an evaluation model based on visual and conceptual similarity. In one version of the system it maximises visual similarity (a value metric) while minimising conceptual similarity (a novelty metric). In another version the target levels for both similarities are user-configurable, meaning it has interactive goals. The Sentient Sketchbook also has interactive goals, providing several sliders and check-boxes for the user to control the weights of its wide variety of value metrics (playability, safety, resource availability, explorability, symmetry, and several forms of balance) plus novelty.

None of the nine systems used a more complex meta-reasoning strategy than metric prioritisation, suggesting that creative meta-search and other forms of goal-based reasoning have yet to be broadly applied in co-creativity research.

**Discussion**

We have suggested a framework for describing interactions within human-computer co-creative systems at three levels.
interaction modalities, interaction styles, and interaction strategies. We have used the framework to analyse nine co-creative systems across four domains – music, poetry, sketching, and dance – and facilitate comparisons in the way they represent, evaluate, reason about and communicate artefacts. The framework illuminates similarities and differences among the systems, and highlights directions for future co-creative systems research.

As interaction modalities are the component of our framework most closely tied to the actual artefacts and their representations, it is challenging to construct a fully domain-agnostic way of describing them. Analysing the interaction modalities used in the example co-creative systems shows a division between those using solely intra-domain information channels and those that supplement with extra-domain information. This allows us to describe systems as being uni-modal or multi-modal in their interaction, although that distinction is less useful in creative domains like music where artefacts are often represented using different modalities. We also used the concept of interaction modalities to identify aspects such as timing differences between systems, as well as restrictions on system feedback.

We analysed the interaction styles used in the sample co-creative systems based on conceptual models, structure of interaction, and structure of contributions. The nine systems were characterised as using ambient, request-based and operation-based models of interaction, as well as being structured to use either simultaneous interaction or turn-taking. This was supplemented with one more level of analysis that characterised the structure of human and computer contributions as additive or iterative. These categories can be easily applied to systems across many creative domains, and suggest the possibility of comparative efficacy studies and the establishment of design patterns and other best practices using this terminology.

Finally, we analysed the co-creative systems for interaction strategies based on metrics, goals, and meta-reasoning capabilities. All the evaluation metrics we found in the example systems were either about artefact value, artefact novelty, or system/user interaction. This taxonomy held across all the creative domains we studied, suggesting its utility as a way of describing co-creative evaluation more generally. The systems we analysed could also be described as following either a static strategy (where their metrics never change) or a dynamic one (where metrics were adapted based on meta-reasoning). The majority had static strategies, and all the dynamic strategies involved some form of metric prioritisation. None of the systems we looked at were able to adapt their evaluation metrics more substantially, nor create new situation-specific metrics based on some higher-level reasoning. This suggests that co-creative systems have yet to leverage some of the research in autonomous creative systems surrounding meta-search and transformational creativity. Co-creative systems that are capable of adapting their evaluation metrics could lead to more intentional and interaction-aware collaboration, linking human–computer co-creativity with the broader discussion on autonomy and meta-creativity (Linkola et al., 2017).

To summarise our discussion, studies of interaction typically ask what is the most effective way to exchange information between a system (a co-creative agent) and its user (its human collaborator) (Bernsen, 2002). In the co-creative context that question is hard to answer, as we lack a rich and general terminology for how concepts, objects, artefacts and more are exchanged. Deconstructing co-creative interaction into modalities, styles, and strategies allows a finer-grained discussion of information flow during human-computer co-creativity. We hope this terminology serves as a useful design tool for exploring different ways that new co-creative systems can be realised. With the framework designers could envision and experiment with different modalities and styles.

Acknowledgments

This research has been funded by the Academy of Finland (decision #311090, Digital Aura) and the Australian Research Council (DP200101059).

References


Bray, L.; Bown, O.; and Carey, B. 2017. How can we deal with the design principle of visibility in highly encapsulated computationally creative systems? In Proc. ICCC, 65–71. ACC.


seer.
Davis, N.; Hsiao, C.-P.; Popova, Y.; and Magerko, B. 2015. An enactive model of creativity for computational collabor-
ation and co-creation. In Creativity in the Digital Age. Springer. 109–133.
Eris, O.; Martelaro, N.; and Badke-Schaub, P. 2014. A comparative analysis of multimodal communication during design sketching in co-located and distributed envi-
François, A. R.; Chew, E.; and Thurmond, D. 2007. Mimi-a musical improvisation system that provides visual feedback to the performer. Technical Report 07-889, Univer-
sity of Southern California CS Department.
Grace, K., and Maher, M. L. 2016. Surprise-triggered refor-
mulation of design goals. In Thirtieth AAAI Conference on Artificial Intelligence.
Granström, B.; House, D.; and Karlsson, I. 2013. Mul-
Hix, D., and Hartson, R. Developing User Interfaces: En-
suring Usability Through Product and Process.
Jordanous, A. 2012. A standardised procedure for evaluat-
ing creative systems: Computational creativity evaluation based on what it is to be creative. Cognitive Computation 4(3):246–279.
Kantosalo, A., and Toivonen, H. 2016. Modes for creative human-computer collaboration: Alternating and task-
Kantosalo, A.; Toivanen, J. M.; Xiao, P.; and Toivonen, H. 2014. From isolation to involvement: Adapting ma-chine creativity software to support human-computer co-
Karimi, P.; Grace, K.; Maher, M. L.; and Davis, N. 2018b. Evaluating creativity in computational co-creative sys-
tems. In Proc. ICCC, 104–111. ACC.
Liapis, A.; Yannakakis, G.; and Togelius, J. 2013. Sen-
Long, D.; Jacob, M.; and Magerko, B. 2019. Designing co-
action” back into uai. ACM Transactions on Computer-
Schuller, B.; Ablaßmeier, M.; Müller, R.; Reifinger, S.; Poitschke, T.; and Rigoll, G. 2006. Speech Commu-
nication and Multimodal Interfaces. Springer Berlin Hei-
delberg. 141–190.
Shneiderman, B. 2007. Creativity support tools: Acceler-
ing discovery and innovation. Communications of the ACM 50(12):20–32.
Weinberg, G.; Blosser, B.; Mallikarjuna, T.; and Ramakrishnan, A. 2009. The creation of a multi-human, multi-robot inter-
active jam session. In NIME, 70–73.
Wiggins, G. A. 2006. Searching for computational creativ-
Young, M., and Bown, O. 2010. Clap-along: A negotia-
tion strategy for creative musical interaction with computa-
Casual Creators in the Wild:  
A Typology of Commercial Generative Creativity Support Tools

Elena Petrovskaya  
Computer Science,  
University of York, York  
ep1167@york.ac.uk

Sebastian Deterding  
Digital Creativity Labs,  
University of York, York  
sebastian.deterding@york.ac.uk

Simon Colton  
School of Electronic Engineering,  
Queen Mary University London.  
SensiLab, Faculty of IT,  
Monash University, Australia  
s.colton@qmul.ac.uk

Abstract
Casual creators are a genre of software that use generative systems to empower autotelic and enjoyable amateur creativity. They have been posited as a unique means of democratising creativity through the support of user exploration via system generativity, yet little is known about what casual creators are actually available to wider audiences. We conducted a qualitative analysis of currently available casual creators on the App Store. We found three categories of interaction techniques across 89 commercially available apps that qualified as casual creators, which we describe in their exploration potential, feedback speed, and user autonomy.

Introduction
Casual creators are a genre of software that playfully facilitates autotelic creativity – for the sake of its own enjoyment (Compton and Mateas 2015). Unlike professional computer-aided design or creativity support systems (such as Photoshop or the Unreal Engine), the primary purpose of casual creators is not to create a professional end product; nor do they require a particular starting level of professional skill, creativity, or even a specific creative intent. Their main aim is affording the enjoyable experience of engaging in creativity, placing focus on the creative process over the creative product. Examples include applications for generating visual imagery, combining basic musical elements into music compositions, or choosing an image from a set to generate similar art. For example, in the app Kaleidoscope Drawing Pad (Bejoy Mobile 2012) (Figure 1), touching the screen generates kaleidoscopic images for amusement.

Casual creators have seen a quick uptake among researchers and practitioners in computational creativity, who have investigated new casual creators across domains, from making live music to writing stories to making whole mobile games (Samuel, Mateas, and Wardrip-Fruin 2016; Kreminski and Wardrip-Fruin 2019; Nelson et al. 2017; Lorway et al. 2019). This is motivated by the promise that casual creators could help democratise creative practice (Nelson et al., 2017) as part of the general rise of amateur (digital) making, which is seen to improve both individual and community wellbeing (Compton 2019, 61-66; Gauntlett 2013).

Compton (2019) proposes a collection of design patterns for casual creators to successfully support autotelic creativity. The key premise of these patterns is to support user exploration by reducing error and offering entertaining feedback, allowing users to approach the creative process with confidence and pride in their work. These patterns work thanks to the incorporation of generativity into the casual creator system, in that the system will alter and enhance the initial user input to produce a more complex output.

However, while generativity is a defining feature of intentionally designed casual creators, Compton (2019) acknowledges that users have adopted a range of systems for autotelic creativity that lack generativity; Compton labels such systems as casual creators ‘in some way’. An example is the app Let’s Create! Pottery (Infinite Dreams 2011), in which the user creates digital pottery in real-time. This system does not involve a generative element, yet is used for autotelic creativity.

Existing work on casual creators has to date focused on such pre-existing exemplars avant la lettre and prototype systems by researchers and artists, with quite small audiences of lab participants, drawing on social networks of researchers and artists. Initial field work on other computational creativity systems (Ackerman and Pérez Y Pérez 2019) has drawn attention to the potential discrepancy between research-based
designs of such systems and their use in the real world, as well as the potential to tap into user feedback to help further design.

If casual creators indeed aim to reduce the barriers of creative practice for broad, general audiences, this raises the question of whether and how casual creators can be designed to actually be broadly accessible and engaging. Do intentionally designed casual creators, with generativity at their core, survive ‘in the wild’ (Rogers and Marshall 2017) in the same way as systems which have been naturally adapted by users to be casual creators ‘in some way’?

One way of approaching this question is to see whether there are already commercially successful casual creators beyond research labs and art exhibitions, and analyse what design characteristics (if any) they have in common. The logic here is that market pressures are likely to have spurred the evolution and spread of designs for casual creation that ‘work’ for broad audiences (Gee 2003). To be sure, commercial availability does not equate public appeal – the majority of games available on Steam or itch.io see little if any uptake. But there is some information to be gleaned from what kinds of applications of a particular genre actually exist in an open marketplace.

In this paper, we report the results from a study following this logic. Specifically, we were interested in two questions: (1) Are there already applications commercially offered to general audiences that can be counted as casual creators? And if so: (2) How are their interactions designed? More precisely, how does their design realise key aspects research has stipulated as essential to supporting casual creation, and can we find specific recurring interaction techniques that do so? Interaction techniques (Hinckley et al. 2014) describe the particular arrangements of input and output that allow a user to perform a particular task.

To answer these questions, we conducted a qualitative review and analysis of creativity support applications commercially available on the Apple App Store in 2019. We found numerous already-existing applications that qualify as casual creators, which clustered into three categories of interaction techniques, and differ in the user autonomy, exploration potential, and feedback they afford. Thus, our paper sketches a first landscape of currently commercially available casual creators, which provides a broader empirical grounding for research. Those looking to develop and deploy their own casual creation systems might benefit from understanding the current landscape, and how their design might be best adapted to fit into this. It is also interesting to see how generative systems are being used in the real world, rather than in content generation for research or artistic purposes, which may provide future insight into user interaction with these systems.

The rest of this paper is organised as follows. We first introduce the concept of casual creators and detail four key design dimensions of casual creators which structure our empirical work: possibility space, feedback, and user/system autonomy. Next, we report our research rationale and design. The Results section details the three types of commercial casual creators we found along the three design dimensions. We close with a discussion of our findings, their limitations and ramifications for future work.

**Background**

Casual creators are designed with the goal of providing a fun and pleasurable experience of the creative process for creative amateurs (Compton 2019). Interaction with casual creators has even been speculated to be playful, placing them on a spectrum with digital games (Compton 2019).

As mentioned above, the key computational enabler underlying the design of casual creators is generativity. A generative method consists of a function which takes initial input and creates a different – often bigger or more complex – output without any additional contribution (Compton, Osborn, and Mateas 2013).

Generative algorithms are often used for art. The key feature of a generative art system is that the user lets a computer system take over some of the decision-making. This is useful for creativity support in several ways: because art is an iterative process, incorporating a computer may help with time-based work by tightening and quickening the iterative cycle. Moreover, it can help with making the decision space smaller and more manageable (Boden and Edmonds 2009).

Generative systems are useful for supporting amateur creativity as they take away some of the responsibility of user creation and add complexity to the final product. Where creativity support tools merely automate or digitally mediate certain steps in the wholesale production of a creative artefact (like copy-pasting or moving a string of notes in a digital music composition), casual creators afford enjoyable amateur creativity by letting users engage with a generative system – e.g. feeding it particular inputs, then observing and selecting outputs from various choices.

Beyond the defining features of generative systems supporting autotelic amateur creativity, Compton and Mateas (2015, Compton 2019) identify a range of design patterns for casual creators. Across these patterns, two key functional aspects or dimensions are repeatedly highlighted and discussed: (1) their possibility space and (2) their fast, entertaining feedback. A further quality discussed or emphasised throughout relates to (3) the relative autonomy of the user over the creative output, also referred to as the power-control trade-off. We will discuss each of these patterns below.

**Explorability: Meaningful-yet-Limited Possibility Space**

Generative systems can be described in terms of their explorability or possibility space: The range of possible
outputs they can produce. As Compton and Mateas (2015; Compton, 2019) stress, for a creativity support tool to ‘feel’ truly creative, their possibility space must be large enough to continually produce novel, surprising outcomes. On the other hand, the almost limitless possibility space of professional creativity support tools can be quickly overwhelming and frustrating for amateur creators. Casual creators offer a comparatively limited possibility space, e.g. by providing starting points, or drastically limiting potential inputs. This guides user exploration of the possibility space and minimises room for error, creative blocks, and anxieties, enabling users to create products which may not have been possible purely on the basis of their own ability.

**Feedback: Fast and Entertaining Evaluations**

As users interact with a generative system underlying a casual creator, they start to build an understanding or mental model of how the system works – how different kinds of input and input dimensions shape the output. Similarly, in creative processes, creators compare their creative vision or idea with its materialised execution.

Compton (2019) argues that feedback in casual creators for any produced artefact should be fast and entertaining. In particular, Compton and Mateas (2015) highlight the theory of reflection-in-action, which argues that people learn from reciprocal interaction with an artefact and then reflecting on said interaction (Schon and Wiggins 1992). Fast and entertaining feedback speeds up learning about the generative system and the creative material one is working with, but also makes the overall experience pleasurable for the users and allows them to feel progress in their creative activity.

**Autonomy: Limited-yet-Meaningful Control**

Compton (2019) puts forward a particular trade-off between power and control as another characteristic of casual creators. Software for creative professionals aims to give the user full, detailed control over the system and end product. Such control is not necessarily essential for amateur creators or autotelic creation: if a user is less concerned with the outcome, they do not need full control over its every last detail. Hence, casual creators shift focus from control to support: they empower the user to produce relatively ‘polished’ outcomes relatively quickly by taking over a large portion of the creative process. This is their power-control trade-off: users sacrifice part of their control over the creative process and product in exchange for increased aesthetic ‘power’ of the overall human-computer system. Transferring some control from the generative system to the user can accelerate the exploration of possibility spaces and learning for the human user, can ensure the end product fits certain aesthetic qualities and requirements, and can make the process more accessible and enjoyable for the amateur user. Again, casual creators achieve this thanks to their generative systems producing rapid and varied outputs, with minimal user input required.

An analytically useful way of translating the trade-off between two aspects (power and control) into one aspect is the relative autonomy of user and computing system, defined as the extent to which an agent has independence over their choices and actions (Barber and Martin 1999). In fact, casual creators have been characterised as mixed-initiative creative interfaces – systems in which human and computer users interact as creative collaborators in a feedback cycle (Deterding et al. 2017; Yannakakis, Liapis, and Alexopoulos 2014). Such systems lie on the midpoint of a spectrum of user and system autonomy between strong computational creativity systems, in which the computer is a fully autonomous creator and humans are merely the audience, and creativity support tools, in which the computer is a tool for the support of fully autonomous human activity. In mixed-initiative systems, neither side has full autonomy over the creative process and outcome. Due to the generative nature of casual creators, user interaction with these creative systems can even be conceptualized as discovery rather than making of creative outputs.

**Study Aims and Method**

As noted, work on casual creators has been chiefly concerned with defining the genre and deriving characteristics and design patterns for ‘good’ casual creators from select case studies that span research prototypes and artistic creations (Compton and Mateas 2015, Compton 2019). Little is known about how many and what kinds of casual creator applications are already broadly available and used ‘in the wild’, what design features they share, and how they realise the three characteristics of casual creators – a limited-yet-meaningful possibility space, limited-yet-meaningful user autonomy, and fast feedback. Specifically, do existing casual creators form some kind of meaningful types, categories, or sub-genres? To answer these questions, we set out to conduct a qualitative review of publicly available casual creators.

**Data collection and sample generation**

To establish a sample of casual creator applications in a reliable and replicable fashion, we broadly followed systematic review procedures, akin to Lister et al. (2014) review of game design elements in mHealth and fitness apps. Between September 26th and October 3rd, 2019, we conducted a search of the Apple UK App Store, running separate searches each for the keywords ‘creative’, ‘create’, ‘creativity’, ‘make’, ‘draw’, ‘art’, and ‘generative’. The Apple App Store was specifically targeted as casual creators are generally developed as mobile applications.

We then defined inclusion/exclusion criteria for counting an application as a ‘casual creator’. We used the definition of casual creators by Compton and Mateas (2015), outlined
above, as our inclusion criterium. The exclusion criterium was “professional creativity tools and apps that don’t involve a generative element”. This criterium was selected because, as noted above, we were interested in analyzing the type of software which can be classified as casual creators by design, rather than applications which have been adapted by users to be casual creators in some way. Generativity in the current paper was conceptualized on the basis of the definition used by Compton and Mateas (2015), who note that generative algorithms ought to produce a ‘wide and interesting space of possible valid artifacts’ (p. 4). Based on this, we included only generative applications which featured an element of randomness in the output, thus being technically able to inspire surprise in the creator.

Our initial search yielded a total 1,121 applications, of which 89 were taken for analysis after applying the inclusion and exclusion criteria.

Coding and evaluation

The applications of the included sample were downloaded onto an iPad and interacted with. This interaction allowed the main interaction technique (Hinckley et al. 2014) of the applications to be determined based on how the user interacted with the system throughout the creative process. Coding and analysis followed qualitative content analysis (Mayring 2004) to identify high-level recurring features, types, or genres. (In the context of qualitative analysis, coding refers to the process of identifying qualitative elements which represent concepts of interest).

Coding was carried out by the same researcher who conducted the initial search and inclusion/exclusion of apps. To ensure the validity of this analysis, coding was initially done at one point, and then repeated after a period of time at a second point. The inter-rater reliability between these time points was calculated as being 0.75, which according to Cohen (1968) constitutes substantial agreement. In situations where there was a disagreement between the two time points, the coding from the second time point was used. Repeated coding at separate time points is a common way of testing inter-coder reliability when only one coder is available (Mackey and Gass 2005).

All codes used were ones which were discovered through the analysis. Three categories naturally emerged from this coding based on shared interaction techniques. The applications in each category were then examined according to the three characteristics of casual creators outlined above: the exploration potential/possibility space, speed of feedback, and user autonomy.

Exploration potential was assessed descriptively based on the size of the possibility space available for exploration within the casual creator: what kind and how many different possible outputs can be produced after one user input?

Feedback speed was assessed by how quickly the user sees any system output after they complete their part of the creative act. Because none of the apps across categories featured any direct delays to system output, assessing feedback speed in time units would not provide a meaningful picture of the differences between apps and categories. Instead, we found it helpful to describe feedback speed in terms of whether the system presents output while or after the user creates.

Lastly, autonomy was assessed using Barber and Martin’s proposed scale of agent autonomy (Barber and Martin 1999), as follows:

- Command-driven: the agent does not make any of its own decisions about how to pursue a goal.
- True consensus: the agent works as a team member, sharing decision control with other agents.
- Local autonomy: the agent makes their decisions alone.
- Supervised autonomy: between command-driven and true consensus.
- Supervisory autonomy: between true consensus and local autonomy.

Results

Table 1 provides a brief summary of the three categories. We then present each category in turn, led by their shared interaction technique. For each category, we give an overview and examples, and then assessments of exploration potential, feedback speed and autonomy.

<table>
<thead>
<tr>
<th>Category</th>
<th>Autonomy</th>
<th>Possibility space</th>
<th>Feedback speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-touch creativity</td>
<td>Supervised</td>
<td>Largest – infinite range.</td>
<td>Moderate</td>
</tr>
<tr>
<td>Vague creation</td>
<td>True consensus</td>
<td>Restricted by user input.</td>
<td>Slowest</td>
</tr>
<tr>
<td>Mutant shopping</td>
<td>True consensus</td>
<td>Restricted by starting variants</td>
<td>Quickest</td>
</tr>
</tbody>
</table>

Table 1. Overview of the categories and their features.

Category 1: One-touch creativity

Interaction technique

The interaction technique of this genre is the repetition of one type of touch gesture, such as tapping, swiping, or flicking the screen to interact with the interface, which leads to the generation of an output.

The user would use one finger to perform a gesture. Touching the screen with a certain gesture yields a generative output which is reflective of this gesture, but otherwise entirely
random – although, in some instances, the user has the option to specify initial parameters, such as colour, meaning the algorithm is parametrized. For example, in *Silk 2* (Vishnevsky 2010) (Figure 2), the user specifies colour and type of line, and the system takes these parameters to create a more complex output. The generative algorithm has stochastic elements, so the output is likely to surprise the user, as they could not fully preconceive the output based on their input. Finally, the generative output is constructive: the user only sees one version of the output, rather than several iterations.

![Image 1](image1.png)

**Figure 2.** Creation in the art system *Silk2*. The user draws and taps the screen, in response to which the system generates aesthetic outputs. The user also has the option to specify some parameters they would like to see in the output, such as line style.

Some one-touch creativity creators involve independent agents whose behavior is influenced by the gesture and makes up the output. For example, in the app `abcdefgijklmnopqrstuvwxyz` (Joerg Piringer 2010), the user directs the movement of agents visualised as letters of the alphabet (with musical accompaniment) (Figure 3).

![Image 2](image2.png)

**Figure 3.** A user swipes the screen in the casual creator `abcdefgijklmnopqrstuvwxyz` to move the letters.

One-touch creativity is the most frequent type of casual creator in our sample, spanning 60% of apps across visual art, music, and text. Notable examples include *Uzu*, a visual interactive light show (Smith 2010), *Figure* (Reason Studios AB 2012), which does so with music by setting up tones and patterns and then creating music simply by sliding a finger, and a series of apps developed by Brian Eno which incorporate both art and music for multimedia creations – e.g. *Trope, Quarto* (*GenerativeMusic.Com | Apps by Brian Eno and Peter Chilvers*’ n.d.).

Visual casual creators of this category often focus on repetition: kaleidoscopes, fractals, and mandalas, e.g. the app *iOrnament* (Science-to-touch 2012), which generates mandalas. Often, this is supplemented with a science theme, such the contextualisation of such art as the creation of particles or molecules.

**Possibility space**

One-touch creativity apps feature the largest possibility space out of all the casual creators. Each gesture yields a stochastically generated element of art from an essentially infinite range.

**User autonomy**

In this way, one-touch creativity casual creators allow only for supervised autonomy: there is an element of choice in how the user executes the gesture and some initial parameters, yet the user does not have much control over the output.

**Feedback speed**

Feedback provision is slower than mutant shopping but faster than vague creation.

### Category 2: Vague creation

**Interaction technique**

Vague creation creators are less common, accounting for only around 25% of our sample. In these applications, users interact primarily through *drawing* something on the screen with their fingers. The user implements some vague or unfinished shape, and the generative system completes this into something advanced. Sometimes this is manifested through the user creating parts, which the program combines to make a whole creation. A distinction between this category and the previous is that the output produced through vague creation is of a complete piece.

The underlying algorithms of creators in this category appear to be parametrized: they produce exactly one deterministic output in response to the user input. Thus, the user experiences less surprise than one-touch creativity.

This type of casual creator spans the creation of visual art and also music. For example, in the app *PendantMaker* (Compton and Mateas 2015), the user makes vague patterns and the app makes a pendant from them. Similarly, in *Scape*, the program recombines musical elements to create new compositions (Figure 4).
User autonomy
There is true consensus in autonomy – while the user does not have full control over the output, the decisions they make on input largely constrain the finished product.

Possibility space
Because the generative algorithm is parametrized, the generative output of this category is therefore based on initial ideas from the user, and the possibility space is also restricted by user input.

Feedback speed
Feedback is slower than the other two categories.

Category 3: Mutant shopping

Interaction Technique
Compton and Mateas (2005) originally identified mutant shopping as a design pattern, which consists of using suggested alternatives to browse the parameter space. We found that it constitutes an interaction technique genre of its own, although this is the least common category, covering only 10% of our sample. This type of casual creator is interacted with by choosing from a selection of starting variants, which serve as the starting parameters. The generate-and-test system then produces a new line-up of variants based on the user’s choice. In some cases, the user has the option to edit the ‘mutant’ before it is reproduced.

Figure 5. Kandinsky.io (Khosravi 2019) – users select their favourite wallpaper variant.

In the casual creator Kandinsky.io (Figure 5), users can generate wallpaper art for their phones, and select their preferred variant out of several others. The element of randomness is quite low in the output, meaning the user is likely to have a clear understanding of what they can expect to see, and is unlikely to experience a high level of surprise. This category is akin to evolutionary art (Romero 2008).

User autonomy
The system and user have true consensus – equally split control over the outcome: the user has autonomy but is constrained by the starting variants.

Possibility space
Exploration potential is very low here: users are only able to create an image by selecting an existing option to ‘mutate’. Given this parametrised nature of the generative algorithm, the possibility space is therefore heavily constrained by the starting variants.

Feedback speed
Feedback provision by the system occurs after every choice the user makes. In this way, feedback is immediate after every choice, and assists with the iterations of the generative system.

Discussion
In this paper, we provided a review and typology of commercially available casual creators. To our knowledge, this is the first broad, data-based assessment of the prevalence of casual creators ‘in the wild’: Compton (2019, p. 6) stipulates that “hundreds (or thousands) of casual creator systems are already part of people’s lives”, but does not provide any evidence for this claim. We found that 89 of 1,121 or about 8% of applications on the Apple App Store findable with search terms associated with artistic creativity qualified as a casual creator, meaning casual creators are indeed available in the wild.

In addition, we employed a qualitative methodology to analyse the main interaction techniques used by said apps, and three distinct categories of casual creators emerged: one-touch creativity, vague creation, and mutant shopping. Again, this presents to our knowledge the first typology of common casual creators. While we cannot claim that these three are the only ‘working’ interaction techniques for casual creators, we do think they may provide insight into what kinds of interaction techniques application developers consider to be proven and working.

We analyzed these categories according to three core characteristics of casual creators: size of the possibility space, speed of feedback, and user autonomy. We found that there were distinct differences in the levels of each characteristic between the three categories. We particularly found that stochastic versus deterministic outputs
An initial overview of the different types of generative algorithms employed in the studied casual creators also lays down some understanding of the optimal level of generativity to benefit users. It is interesting that the most widely available category of casual creators – one-touch creativity – is also the type of casual creator with the biggest potential to surprise the user with the output. The implications of surprise in generative art creation are something we would like to consider in future work.

Limitations
The most pertinent limitation to this work is that it is based on a limited sample, assessing only applications available on one particular digital marketplace (the Apple App store) at one point in time (2019). Also, the number of available applications says nothing about their actual uptake in terms of e.g. number of installs, users, or time spent on app.

Furthermore, coding and analysis were conducted qualitatively by a single researcher. While re-coding at a separate time point ensured some degree of inter-coder reliability, we acknowledge that re-analysis with several coders and a pre-defined coding handbook would produce more reliable results. This holds especially for categorizing applications in terms of what experiences of autotelic creativity they afford.

Future work
This work is a first step in assessing the evolving landscape of creativity support tools and casual creators. We lay down foundations of how different levels of generativity between casual creators may affect the way users interact with the systems, and in our future work we will explore this through a more user-centered approach.

Next steps should also look further into the general user experience of casual creators, and how the described design features of possibility space size, feedback speed, and user autonomy contribute to this experience. We are already working on the construction of a grounded theory of what might motivate the construction and sustainment of user engagement with these apps. Initial findings show the distinction between casual creators as interfaces and casual creation as a mindset. We are interested in seeing what the contribution to this mindset is of the various features outlined in the current work, and how the design of casual creators can be further improved to facilitate autotelic creativity.

We hope that this paper helps build an initial understanding of the user side of casual creators, and patterns of interaction with casual creators in the wild.

Conclusion
This study conducted a qualitative review of applications designed to support casual creativity on the AppStore and found 89 commercially available apps that fit the definition of casual creators. Qualitative analysis revealed three distinct categories of interaction techniques. Those categories were analyzed based on core design characteristics of casual creators: size of the possibility space, feedback speed, and the level of user autonomy in the creative act. The categories of casual creators differed in their levels of these characteristics and the level of generativity provided by the systems. This work outlines an ‘in the wild’ landscape of casual creators, and points towards directions for further work into the user experience of amateur digital creativity, as well as support for those looking to commercially release their own casual creators.

Acknowledgements
This work was supported by the EPSRC Centre for Doctoral Training in Intelligent Games & Games Intelligence (IGGI) [EP/L015846/1 and EP/S022325/1] and the Digital Creativity Labs (digitalcreativity.ac.uk), jointly funded by EPSRC/AHRC/Innovate UK under grant no. EP/M023265/1.

References
Davis, Nicholas Mark, Chih-Pin Hsiao, Kunwar Yashraj Singh, and Brian Magerko. 2016. ‘Co-Creative Drawing Agent with Object Recognition’. In Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference.
Conference Extended Abstracts on Human Factors in Computing Systems, 628–635. ACM.


Lister, Cameron, Joshua H. West, Ben Cannon, Tyler Sax, and David Brodegard. 2014. Just a Fad? Gamification in Health and Fitness Apps. JMIR Serious Games 2 (2), E9.


Reason Studios AB. 2012. Figure. Mobile application. https://www.reasonstudios.com/mobile-apps


Exploring Crowd Co-creation Scenarios for Sketches

Devi Parikh\textsuperscript{1,2} and C. Lawrence Zitnick\textsuperscript{1}
\textsuperscript{1}Facebook AI Research
\textsuperscript{2}Georgia Tech
parikh@gatech.edu zitnick@fb.com

Abstract

As a first step towards studying the ability of human crowds and machines to effectively co-create, we explore several human-only collaborative co-creation scenarios. The goal in each scenario is to create a digital sketch using a simple web interface. We find that settings in which multiple humans iteratively add strokes and vote on the best additions result in sketches with the highest creativity (combination of value and novelty). Lack of collaboration leads to a higher variance in quality and lower novelty or surprise. Collaboration without voting leads to high novelty but low quality.

Introduction

How can one best collaborate with humans in a creative process? Insights towards this can inform what roles machines can (or should not) play when co-creating with humans.

Specifically, we consider a scenario where agents take turns collaboratively drawing a sketch on a simple web interface (Figure 1). During each iteration, multiple agents propose strokes to add to the sketch. Agents then vote on the proposals, and the preferred set of strokes is added to the sketch. This process is repeated for a fixed number of iterations to create a final sketch.

The roles of creating stroke proposals and voting could each be fulfilled by either humans (H) or machines (M). Borrowing terminology from Generative Adversarial Networks (Goodfellow et al. 2014), we can call the former role a generator (G), and the latter a discriminator (D). This allows for \(4 \times \{G,D\}\) co-creation scenarios. Further, different individuals could play the role of generators/discriminators across iterations, leading to crowd co-creation.

In this work, as a step towards human-machine co-creation, we study various human-human crowd co-creation scenarios. In the first, Individual, a single human creates the entire sketch (no discriminator D, and no crowd). Second, in Collaborative the sketch is generated by multiple human agents (crowd) iteratively taking turns adding strokes. That is, all the agents act as generators G and there is no voting or discriminator D. The third, Collaborative + voting, is where multiple human agents (crowd) propose new strokes at each iteration. Another set of human agents (discriminators) vote on which set of strokes to add to the sketch. Finally, we explore Individual with collaborative prompts, for which the crowd is involved indirectly. A single human creates the entire sketch, but by following text prompts that describe the evolution of a sketch that was created in the Collaborative scenario.

We evaluate the qualitative difference between the sketches produced via these four scenarios. We find that the collaborative setting with a voting mechanism (Collaborative + voting) leads to sketches that are rated by human subjects as most creative (and are preferred along a variety of other dimensions). The lack of either one of these components results in less creative sketches: Individual sketches have decent quality (value) but low novelty, while Collaborative sketches have high novelty but low value. Individual with collaborative prompts results in high novelty but even worse quality. Overall, among these four scenarios, Collaborative + voting best hits the sweet spot for creativity: combination of value and novelty (Boden 1992).

Figure 1: As a first step towards human-machine co-creation, we explore human-human collaboration for creating digital sketches on a simple web interface shown above. Video: https://youtu.be/9fikuKYPdO

---

\textsuperscript{1}Facebook AI Research
\textsuperscript{2}Georgia Tech
parikh@gatech.edu zitnick@fb.com

Proceedings of the 11th International Conference on Computational Creativity (ICCC'20)
ISBN: 978-989-54160-2-8

73
Figure 2: Few iterations of a sketch being created in the **Collaborative + voting** scenario. Once a parent sketch gets five children, it gets selected as the next iteration of the sketch (black outline), and the five children become the parents for the next iteration. Temporal visualization: https://youtu.be/JQmGALAhMU.

### Related Work

In the context of crowd-based sketching, (Tuite and Smith 2012) analyze user actions and large-scale behavior patterns in 50k sketches from Sketch-a-bit, a collaborative mobile drawing application. Different from our incremental contribution and voting mechanisms, (Yu and Nickerson 2011) and (Gingold et al. 2012) explore combination and averaging of sketches respectively as collaboration strategies.

Several AI systems have been trained to recognize sketches (e.g., models trained on The Quick, Draw! Dataset¹). These may form useful building blocks for the next stages of our work. However, as seen in Figure 3, our sketches tend to be complex scenes and often abstract as opposed to concrete individual objects, which has been the focus of most existing work in automatic sketch recognition. There is also work on generating images based on sketches. Central to their study is the back and forth interaction (dialog) between the human and machine as they take turns. Our work is focussed on a crowd setting where no two individuals contribute to a sketch. The first subject sees a blank canvas and adds strokes. Every subsequent subject is shown the partial sketch and asked to add to it. They cannot undo strokes from earlier contributors. The prompt is “Let’s collectively create a beautiful, detailed, coherent painting!”.

### Co-creation Scenarios

We explore four scenarios for collaborative human-human sketch co-creation. In every scenario, the sketch starts with a blank canvas. During each iteration, a limited number of strokes may be added. The limit roughly corresponds to five medium-thickness strokes spanning the width of the canvas. 30 iterations are used to create each sketch. Unless stated otherwise, we collected 20 sketches for each scenario. All our studies were conducted on Amazon Mechanical Turk. Subjects can not submit their work till they have contributed the required amount of strokes to the canvas.

**Individual.** The entire sketch is created by a single individual. That is, a single human agent adds all 30 iterations of strokes to “Create a beautiful, detailed, coherent painting!”.

**Collaborative.** A different human agent contributes strokes for each iteration of the sketch. That is, 30 unique individuals contribute to a sketch. The first subject sees a blank canvas and adds strokes. Every subsequent subject is shown the partial sketch and asked to add to it. They cannot undo strokes from earlier contributors. The prompt is “Let’s collectively create a beautiful, detailed, coherent painting!”.

Subjects are given the additional instruction to consider the kind of painting being created and the stage of the painting when deciding upon which strokes to draw.

**Collaborative + voting.** Each subject contributes strokes to a sketch of their choosing from a set of five starting sketch variations. We refer to the chosen starting sketch as a parent, and the sketch created by a subject as the chosen sketch’s child. During each iteration, sketches are gathered until a parent is selected five times. Its children then replace the current five parents and the process is repeated. Children of parents selected less than five times are discarded. See Figure 2 and https://youtu.be/JQmGALAhMU.

This voting strategy allows for the most promising versions of a sketch to go forward. This scenario is robust to the strokes added by any one individual. Of course, it is also significantly more “expensive”. In the best case scenario where a single parent gets all 5 children and none of the other parents get a child, it takes 5 times the amount of strokes to create a sketch compared to Collaborative. In the worst case, all 5 parents get 4 children each before a parent

¹ https://github.com/googlecreativelab/quickdraw-dataset
Individual with collaborative prompts

Collaborative + voting

Collaborative

Individual

Figure 3: Example sketches from four co-creation scenarios along with differences identified by human subjects between sketches from pairs of scenarios. **Collaborative + voting** involves ~12.5 times the individuals, and so was run for 20 instead of 30 iterations. For comparison, **Collaborative** sketches are also shown at 20 iterations.

Figure 4: Example prompts used in the **Individual with collaborative prompts** scenario.

Figure 5: Evolution of example sketches in the **Collaborative** scenario. Left: Focus of the sketch shifts from the house to the cat in the rain outside the house. Right: Faced with seemingly incoherent strokes, subjects emphasize structure they see in it so subsequent subjects can add to it.

gets a fifth child. This would result in 21 times the number of strokes. In practice we found this factor to be about 12.5 times. Given the increased cost, we reduced the number of iterations in this scenario to 20 (as opposed to 30). On average, 250 unique individuals contribute to a single sketch.

**Individual with collaborative prompts.** A single individual creates an entire sketch using instructive text prompts provided at each iteration. The individual is instructed to follow the prompts when drawing. The text prompts are generated by asking another individual to describe what changed in a sketch from one iteration to the next in the **Collaborative** scenario. All text prompts for a sketch are written by a single individual. This is an interesting hybrid of having a single creator, but being guided through prompts that describe the evolution of a sketch as created by 30 unique individuals. We collected three sets of text descriptions for each of the 20 **Collaborative** sketches. This resulted in a total of 60 **Individual with collaborative prompts** sketches. In our evaluation, we consider 20 sketches (randomly picking 1 out of the set of 3). See Figure 4 for example prompts.

**Evaluation**

Example sketches from these scenarios are shown in Figure 3. Before we discuss properties of the final sketch, it is worth considering the evolution of a sketch as it is being created. **Collaborative** sketches evolve in several interesting ways: what seems like the main subject of a sketch changes in a few iterations (Figure 5, left), given seemingly incoherent strokes, subsequent subjects try and emphasize regions that could lead to meaningful structures in the sketch for future subjects to build on (Figure 5, right), and subjects use the color white or other strategies to try and cover parts of the sketch they think are contributing negatively to it.

To assess the qualitative differences between sketches produced from the 4 scenarios, we created a collage of 20 sketches from each scenario (at 20 iterations for **Collaborative** and **Collaborative + voting**, 30 for the rest). We showed pairs of collages to subjects on Amazon Mechanical Turk and asked them to describe differences that stood out. Snippets from subjects’ responses are shown in Figure 3.

For a quantitative evaluation, we showed subjects pairs of sketches from two different scenarios (i,j). Each subject picked which sketch they prefer along 12 axes. Every pair was evaluated by 5 subjects resulting in 144,000 assessments: 20 (sketches from scenario i) × 20 (sketches from scenario j) × 6 (pairs of scenarios) × 12 (axes) × 5 (subjects per sketch-pair). The 12 axes were: which painting (1) seems more strange / unusual / different than typical paintings? (2) is a better painting? (3) do you like looking at more? (4) is more creative? (5) is more interesting? (6) is more original? (7) took more skill? (8) is made by an artist more likely to be an adult? (9) would you pay more for? (10) are you more likely to put up in your home? (11) is more likely to be in an art museum? (12) would you be more proud to have made yourself? Some of these axes (e.g., originality, novelty, skill) are from (van der Velde et al. 2015).
The percentage of times each scenario was picked over a competing scenario is shown in Figure 6. For 11 out of the 12 axes, including creativity, Collaborative + voting is preferred. Collaborative + voting scores well for both novelty (unusual) and quality (better, look), which we hypothesize increases its perceived creativity. Individual is rated well for quality but scores poorly on novelty. Across 11 out of the 12 axes, Individual has high variance due to differences in skill and motivation of individuals creating the sketches. Collaborative scores well on novelty, but worse on quality. Individual with collaborative prompts does poorly across all axes except for unusual. Of all scenarios, Collaborative + voting falls in the sweet spot for maximizing creativity (combination of value and novelty).

Discussion

In what way may a machine best contribute to the collaborative creation of a sketch? It is often the case that humans may not be good at generating strokes, but can tell if a sketch looks good or not. This may suggest using machines to generate candidate strokes and having humans vote on which version should proceed next. The machine may also contribute in a manner similar to the humans in our fourth scenario, i.e., the machine could generate textual prompts as a human draws a sketch. The prompter can have different “personalities” based on whether it is trained on sketches a human draws a sketch. The prompter can have different “personalities” based on whether it is trained on sketches individual (coherent), Collaborative (rich but chaotic) or Collaborative + voting (rich with subtle details and coherent). Humans and machine can generate strokes as a team, either in co-painting scenarios as in (Cabanues et al. 2019), or where the machine provides some visual guidance as in (Lee, Zitnick, and Cohen 2011) or via suggestions for where to draw, what colors to use, etc. as explored in (Oh et al. 2018). We can also train a machine to be a discriminator: given a few different strokes from a human, select which stroke should be added to the sketch next.

All our sketches started with a blank canvas. We could instead start sketches with a prompt (subject of the sketch, adjective describing a desired property of the sketch, a picture to be used as inspiration for the sketch, etc.), and have this prompt persist across iterations (or not).

Our collaboration scenarios can be enriched by allowing for repeated interactions between the individuals contributing to the sketch, creating opportunities for a dialog.

It is interesting to consider ideas of ownership in the context of crowd co-creation. While no one individual may feel a complete sense of ownership of the final piece, crowd collaboration may lead to a sense of community and the satisfaction of contributing to a common cause. Finally, while our motivation was human-machine co-creation, studying human-human collaboration in general is, obviously, important and interesting in and of itself. Collaborative creative endeavors may be a fertile ground for such explorations.

References

Yu, L., and Nickerson, J. V. 2011. Cooks or Cobblers? Crowd Creativity through Combination. In CHI.
A Climate Change Educational Creator

Jun Chang and Margareta Ackerman
Department of Computer Science and Engineering
Santa Clara University, Santa Clara, CA

Abstract
Casual creators offer an enjoyable and readily accessible creative experience by enabling a safe and easy exploration of a creative space. This explorative and intrinsically motivating process lends itself to education applications, giving rise to a new category of casual creators, which we call Educational Creators. We illustrate this concept through EarthMood, an educational creator for deepening students’ understanding of Climate Change. We demonstrate EarthMood through historical data, showing the deterioration of the planet, as well as on recent data, illustrating an improvement in climate due to COVID-19. The remotely-accessible nature of educational creators makes them applicable to both traditional and remote learning settings.

Introduction
Recently, there is a growing discussion on “why, and for whom, do we create CC systems?” (Cheatley, Moncur, and Pease 2019). A wide spectrum of Computational Creativity systems offer value in a variety of ways. For example, an autonomous system may provide value through its artifacts. On the other hand, a co-creative system may enable an artist to more efficiently or effectively engage in a creative process.

Casual creators represent a distinct type of creative systems, allowing a user to easily explore a creative space through playful interaction (ex. moving a cursor on the screen, or controlling sliders). Bringing the question from (Cheatley, Moncur, and Pease 2019) to casual creators, we ask: “Why, and for whom, do we make Casual Creators?”

Casual creators are often made for their autotelic creativity (Compton and Mateas 2015), which is the inherent and intrinsic satisfaction of a creative activity for its own sake, without necessarily having any extrinsic aims. We propose that the intrinsic satisfaction of engaging with casual creators need not preclude additional utility. In fact, since their inception, casual creators carried additional, albeit perhaps secondary, aims. For instance, PendantMaker, an early illustration of a casual creator (Compton and Mateas 2015), enables the creation of pendants that can be subsequently 3D printed and worn. In addition to utilizing resultant creative artifacts, what other goals may be accomplished through casual creators?

Educational creators may offer a particularly engaging form of remote learning. They can be made available online and require minimal to no teacher supervision. As with casual creators, engagement with an Educational creator puts the user (or, student) in the driver’s seat. Just as a casual creator allows users to take creative ownership, educational creators have the potential to support intrinsic motivation by offering an inherently enjoyable and user-directed interaction.

In this paper, we illustrate the concept of a casual creator by introducing EarthMood, a system that deepens a student’s understanding of climate change. EarthMood visualizes climate change through simple, playful visualizations. The user can adjust sliders representing critical elements such as air pollution (AQI) and immediately see the impact of such changes. By offering an inviting and playful interaction, EarthMood is suitable for giving school-aged children
Figure (2) Some of the sliders available to the user to control EarthMood’s visualization of climate change. Such controls allow the learner to easily explore the space of creative visualizations corresponding to the Earth’s climate condition.

a deeper understanding into the impact of climate change, and as such raise awareness of this critical issue.

This paper illustrates the potential of educational creators through one particular system and application. We believe that educational creators have the potential to support education across a wide array of domains by integrating casual creation into the learning of complex topics.

EarthMood

EarthMood is a casual creator that supports the understanding of Climate Change by turning data into an artistic and relatable visual representation. The user engages with the system by controlling how much pollution is released and seeing its impact on the earth in real time, through creative visualization of the data. Using data such as CO2 level in the atmosphere, global temperature, and sea levels, a digital earth is animated to restore empathy to our planet and facilitate a meaningful learning experience. Controlling such conditions through sliders (as depicted in Figure 2) lets the user explore the creative space of visualizations representing various climate conditions, and as such gain an understanding of how human activity impacts the planet.

The visualizations are done entirely through rectangular shapes, resulting in an artistic yet approachable style. Users engage with EarthMood in its animated, live form. See Figure 3 for static illustrations from EarthMood, showing the planet in various stages of pollution.

To better understand the visuals, consider, for instance, Figure 3(a). The green zone represents the condition of the earth, while its blue surrounding captures the health of the planet’s water bodies. When using EarthMood, the rectangular elements are in a constant state of motion and reflect the user-selected settings, leading to an engaging and responsive experience. The more representative video format, along with two examples, is discussed in the “Video” subsection below.

Data visualization

EarthMood uses climate change data such as CO2 ppm, ocean pollution, global temperature, species diversity and

(a) A frame from EarthMood’s representation of earth’s climate with healthy biodiversity and no visible pollution. The green zone represents the condition of the earth, while its blue surrounding captures the health of the planet’s water bodies.

(b) Ocean pollution is introduced into the system as black squares start appearing in the ocean. Rising sea levels cause the landmass to shrink.

(c) Earth becomes more crowded as population increases. Opaque squares start to obscure the Earth as air becomes more polluted.

(d) The Earth is barely visible under the dense atmosphere filled with air pollution and the waters are continuing to get polluted.

Figure (3) An illustration of climate deterioration, from healthy (a) to polluted (d). Best viewed in color.
more to visualize the impact of human activity on our planet. Unlike tables and graphs, which fail to evoke emotion, and as such are often ineffective at communicating the devastating impact of climate change, EarthMood aims to create impactful and engaging moving visuals.

The resulting visuals utilize vibrant colors that gradually gets duller as the negative human impact increases. From a series of different simulations shown in Figure 3, we can easily grasp the differences between a polluted Earth versus a clean Earth. These bright visualizations and the radical differences between them aim to elicit emotion in the viewer which turns a “meaningless” statistic into readily relatable images. The student is able to easily manipulate the data through sliders to see the impact of their actions immediately visualized in the artwork.

Features
Below is a list of features used to create EarthMood’s moving visualizations:

- **Air Quality Index**: Increases the amount of smoke cloud squares covering the Earth as air quality drops.
- **Endangered Species**: Decreases the brightness of the squares appearing on the Earth as more species become endangered.
- **Carbon Dioxide PPM**: Decreases the opacity of the smoke cloud squares covering the Earth as levels rise.
- **Global Sea Level**: Changes the ratio between the Earth and the ocean, shrinking the Earth as sea levels rise.
- **Global Average Temperature**: Shifts the color of the smoke clouds covering the Earth between blue and red.
- **Ocean Plastic Pollution**: Shifts the color of the ocean from blue to green as ocean pollution grows.
- **Global Population**: Increases the number of squares appearing on Earth as population grows.
- **Earthquakes**: Shakes the screen when an earthquake occurs.

Videos
The interaction with EarthMood is more accurately captured through video than static images. The following video demonstrates EarthMood in action, visualizing climate data from 2000 to 2018:

https://youtu.be/rbrTKzkxyt20


In addition to visualizing historical data, each parameter of EarthMood can be manually adjusted for the user’s experimentation. The following video demonstrates this aspect:

https://youtu.be/UveQMK13_m0

Implementation
EarthMood is implemented in Java. It uses a parser to take in data stored in CSV format and generates the images accordingly. The square generation uses a pseudo-random function to determine the position, color, size, and duration of the squares within specified ranges. For the animation, the gpdraw library is used to paint the shapes onto the screen. Multiple threads are implemented to ensure no lag in the animation as the user is changing the parameters. The user control is done with a combination of gpdraw library and Java swing.

Case Study: New Delhi and COVID-19
New Delhi traditionally suffers from air pollution due to high population density. This leads to health risks including heart attacks, asthma, bronchitis and lung cancer (National Geographic 2020). However, during the COVID-19 crisis, the Ministry of Environment reported a significant drop in air pollution in India’s capital (Central Pollution Control Board, Ministry of Environment, Forest and Climate Change, Government of India 2020).

Poor air quality comes from a multitude of sources such as motor vehicle emissions and coal-based power plants. Due to the stay at home order, many sources of air pollution have been dramatically reduced. This is reflected in the visualizations from EarthMood.

Figure 4 (a) and (b) depict the state of Delhi’s climate in November 2019 and April 2020, respectively. A quick
glance at the two figures reveals the substantial climate improvement that took place over this short period of time. In particular, we note the much improved air quality in Figure 4(b), depicted through fewer smoke clouds, revealing the colorful squares beneath.

**Discussion**

EarthMood aims to represent our planet in a dynamic and relatable way in order to elicit a sense of kinship between the viewer and the earth. Using data collected over the past century, an interactive learning experience is created with the aim of giving a (potentially young) user deeper insight into the significance of climate change.

Letting students instantly grasp the impact of their slider manipulations on the earth’s climate through simple visualizations is central to an easy and impactful learning experience (this relates to the instant feedback design pattern of casual creators (Compton and Mateas 2015)). A quick glance at the generated images is sufficient to infer the shocking impact of, for example, a small change on the CO2 slider. This self-guided and responsive educational creator aims to offer an engrossing and effective learning experience.

Preliminary analysis suggests that EarthMood evokes feelings in its viewers and deepens their understanding of the impact of human activity on our planet. Further study is needed to evaluate the impact and educational value of EarthMood, particularly in the context of remote learning, and to identify whether this project can have long-term impact on human behavior.

EarthMood’s educational value can be extended through, for instance, additional sliders that would allow a student to represent human activity, such as recycling behaviour, the impact of which would be reflected in the visualizations. Furthermore, additional visualization styles can be explored, and complemented with generative music to reflect the state of the planet.

This work initiates the study of educational creators, along with an example of one such system. Additional examples of education creators, as well as a comparison to other educational systems related to Computational Creativity (ex. the educational value of Mixed Initiate Creative Interfaces (Deterding et al. 2017)) are left to future work.

Educational creators can offer impactful learning experiences by combining the inherent appeal of artistic artifacts with the engaging experience arising from participation in a creative process. The artifact may, for example, span visual art, music, poetry or virtual reality experiences. Simple, playful artifacts or more complex ones may be used. We conclude with a summary of the main characteristics of educational creators:

- **Intrinsically motivating**: Based upon the autotelic creativity of casual creators, the joyful experience of creating with an educational creator is likely to give rise to intrinsic motivation.
- **Explorative**: Casual creators support an enjoyable exploration of a creative space. Educational creators place the learners in the driver’s seat and are likely to evoke students’ curiosity.
- **Learning through play**: The playful experience offered through educational creators makes this a suitable approach for young learners.
- **Remotely accessible**: Educational creators can be made widely accessible through the web, making them easy to incorporate into remote learning curricula.

We believe that educational creators can be valuable across a wide array of disciplines, particularly where students find it challenging to connect with a subject matter through traditional educational approaches.

**References**


Arny: A Co-Creative System Design based on Emotional Feedback

Sarah Abdellahi, Mary Lou Maher, Safat Siddique
Software & Information Systems Department
University of North Carolina Charlotte
Charlotte, NC, USA
sabdella, m.maher, ssddiq6@uncc.edu

Abstract
In human-human collaboration, gesture, verbal communications, and emotional responses are among the communication strategies that shape the interactions between collaborators to negotiate the contributions. Emotional feedback allows human collaborators to passively communicate their stance about the experience and convey their perception of the process without distracting the flow of the task. In human-human co-creative collaboration, participants interact and contribute to the task based on their perception of the collaboration over time. In human-AI co-creativity, perceiving the cognitive and emotional state of the user in order to determine the dynamics of collaboration and decide what the agent should contribute to the artifact are two primary challenges of building effective co-creative Artificial Intelligence systems. This paper addresses these challenges with the design of a co-creative agent that can sense the user’s emotional state through facial expression.

Introduction
Numerous researchers are interested in designing co-creative systems that are capable of creative collaboration with humans due to the wide range of creative engagement benefits present in different areas from healthcare to education and industry (Sanders & Stappers, 2008; Makhaeva et al, 2016). However, there are currently two primary challenges to building effective co-creative systems: 1) determining the interaction dynamics of the collaboration, e.g. whether the system should lead, follow, wait, and 2) determining what the agent should contribute to the shared creative product and why. This paper contributes to the research on mixed initiative creative interfaces (Deterding et al 2017) by including emotion feedback in the interface design. Human collaboration negotiates both factors through verbal and non-verbal social cues. Humans communicate their stances about these matters through conscious and unconscious body gestures, facial expressions and verbal feedback. Such feedback reflects the human feelings and the value judgments of each stage of the collaborative process and guides the interaction and contribution. However, current designs of co-creative systems lack such negotiation mechanisms. Designing a feedback mechanism that effectively informs the co-creative agent how it should behave to increase creative engagement and fluid interaction is a challenge addressed by a system design introduced in this paper.

Emotions align with different human cognitive states and allow humans to reflect and communicate with their collaborative partners in order to calibrate the collaboration dynamics and their behavior. In a computational setting, emotions are an ideal candidate for feedback since they are passive, meaning that the user does not need to explicitly click or say anything. Unlike other negotiation methods such as voting buttons and verbal feedback, the passive characteristic of emotional feedback allows negotiation over the matters to happen without distracting the flow of the process. Moreover, passive emotional feedback does not require the user to learn or get used to any new method of communicating feedback as is required by methods such as verbal feedback. We explore the use of emotion as a feedback mechanism to modulate the interaction dynamics and behavior of co-creative agents.

Background
Considering the critical role of emotion in human interactions, the field of Human-Computer Interaction seeks to make human system interactions more spontaneous and humanlike by including emotions in the dynamic. In human-human interactions “Emotional expressions are crucial to development and regulation of interpersonal relationships” (Ekman, 1999). Studies have shown individuals with facial paralysis experience great levels of difficulty developing and maintaining even casual relationships as they are incapable of expressing emotions effectively. This observation about human interactions inspired the current trend of considering affect in systems design.

When it comes to collaborative systems however, consideration of affect is much more than a user expectation and desire. In a collaborative system when the user and the system contribute to a shared task, consideration of affect has a significant influence on the interaction between the
participants and consequently the main sensemaking flows of the collaboration. Emotions elicited by stimuli events trigger responses in the participants and allow them to adapt to the collaboration (Scherer, 2005; Sawyer, 2014).

Kellas and Trees (2005) refer to two distinct sense-making processes in open-ended improvisational interaction: 1) functional sense-making that determines the content generated for a particular turn, e.g., choosing to draw a house or pattern, or choosing which words to say, and 2) interactional sense-making that structures and maintains how the interaction is unfolding through time i.e. the interaction dynamics, such as turn taking, turn length, and the overall rhythm of interaction. Participatory sense-making occurs when there is a mutual co-regulation of these two sense-making processes between multiple participants, i.e. both participants are adapting their responses to each other and working to maintain an engaging interaction dynamic that supports the mutual exchange (De Jaegher & Di Paolo, 2007).

When participatory sense-making occurs, the interaction can take unpredictable paths and new ideas can emerge by traversing through new conceptual spaces and generating responses to unpredictable queries. In human collaboration, collaborators naturally co-regulate their sense-making processes through awareness of their collaborator's judgment of their contribution at each point of time. Awareness of collaborator’s emotions during a collaboration allows the participants to validate their actions from their collaborator’s point of view and use this awareness to proceed with the participatory sensemaking (Eligo et al, 2012). The meaning structures built by the interactional sense-making process guide the interaction forward by suggesting what can be added next given the history of the interaction. Also, interaction patterns are developed that circumscribe the type and amount of content to be generated at a given time. For example, when getting to know each other, people often employ a question and answer interaction pattern that suggests when each person should ask another question to keep the interaction moving. The same concept of having a pattern can be true for collaborative drawing – interaction dynamic patterns emerge such as call and response, mimicry, mutual building, antagonism, and transformation.

Once an interaction pattern is established through awareness of affect and context, cognitive resources can be turned from interactional sense-making to functional sense-making, and the participant can focus solely on generating a response to their partner in line with the latest interaction pattern being employed rather than generating a new contribution from scratch. This process can repeat and observed changes in the affect or shared product during the collaboration could direct the participant to choose to re-engage in interactional sensemaking to come up with a new way of interacting and establish a new interaction pattern.

**Arny: Co-creative Interaction with Emotion Recognition**

Arny is a co-creative drawing system based on emotional feedback. The initial idea of considering emotional feedback in AI-based co-creativity is rooted in our observations during a previous study of Human-Human creative collaboration. During that study, individuals were paired with a facilitator from our research team to collaborate on a set of open-ended drawing tasks. After the drawing tasks the participants reflect on the experience during a retrospective protocol (Abdellah et al, 2019). Thematic analysis of that case study shows that different individuals’ responses to similar drawing contribution types that were explored by our facilitator varied in many ways. However, despite this difference all these participants reported a collaboration strategy structured around reflecting on the facilitator actions outside the drawing canvas through verbal feedback and body language. This behavior pattern is in line with the findings of Sawyer (2014) about how human interactions during the collaboration impacts the collaborator’s contribution. Moreover, we perceived that individuals with a stronger focus on interactions and communicating their perception and feedback reported a higher level of satisfaction of the collaboration quality during our study. These observations inspired us to consider a Human-AI collaboration model that considered not only the collaborators’ contribution to the task, but also a method of interaction to communicate perceptions and expectations similar to those in Human-Human collaboration. We decided to utilize the communication of affect through facial expression to address the need for a channel of passive interaction between the user and the AI system.

The current design of Arny, as presented here, is a result of multiple phases of iteration and evaluation.

**Role of Affect in Arny’s System Design**

In order to design an affective system, some fundamental aspects have to be decided on. The first and main decision to be made is what is the “role of affect” in the system design. Consideration of affect in system design can refer to recognizing user affect, adapting to a user’s affective state, generating affective states within an agent or a combination of these options (Hudlicka, 2003). The interpretation addressed by Arny is recognizing the user’s affect and adapting to the user’s emotional state.

While a typical AI-based co-creative system relies on functional sense-making in order to collaborate with their human colleagues, Arny follows a model shown in Figure 1 to incorporate the user’s emotion in the system design and include the interactional sensemaking component as part of its sensemaking cycle.

Since the role of affect in the design of Arny is to investigate the user emotional state and adapt to it, another fundamental aspect to investigate in Arny’s design is affect recognition. The emotion recognition method in Arny uses the facial expression method of emotion detection as it provides a sufficient level of accuracy for the purpose of this research. Using this method, emotions can be measured in real time and without distracting the user or interrupting the user-system interactions. Currently Arny uses Affectiva facial expression recognition toolkit, a 3rd party tool available as an add-on to the IMotion biometrics evaluation package.
Affectiva captures the participant’s facial expressions in real time using a simple webcam (“Affectiva”, 2020).

Figure 1. Sensemaking model in Arny

Arny perceives and buffers the participant’s emotional responses to the contributions performed by the system and uses them in its decision-making process. The emotional feedback perceived by Arny is characterized by the valence and engagement values using the data provided by Affectiva. Arny categorizes the valence values as positive valance, negative valance or neutral and the engagement values as low, or medium-high engagement. The combination of these valence and engagement categories are used by Arny to interpret the user’s value judgement of the contributions from themselves and Arny as a basis for choosing Arny’s next contribution.

Interaction Model

To design Arny we started with a primary version of an interaction model influenced by our previous observations of human creative collaborations. Preliminary evaluations of this model then guided us to suggest the current version of an interaction model for a co-creative collaboration context. While these evaluations are beyond the scope of this paper, we present them in Abdellahi et al (2020).

Arny’s interaction model includes two basic components, a “collaboration model” and a set of “response rules”. The collaboration model identifies the basic rules of the collaboration in addition to an overview of the feedback types used by Arny. The response rules go deeper and define how Arny responds to each specific feedback type with consideration of the collaboration rules.

The collaboration model followed by Arny aims to keep the user’s affect close to positive throughout the process by suggesting a next action type that is least likely to trigger a negative future affect. For this reason, it is important for Arny to be able to identify the contribution that is the source of a triggered emotion while interacting with the user.

Emotions are generally elicited by two types of stimuli: 1) External stimuli, when outside events trigger emotion, such as natural causes or behavior of other people; and 2) Internal stimuli, when one’s own behavior can be the event that triggers the emotion, such as with pride or shame (Scherer, 2005). Emotions experienced during a collaboration could be triggered by actions of the collaborator, collaboration environment, or by one’s perception of their own contributions.

To allow Arny to identify the source of emotions, the collaboration model follows a turn-taking pattern for collaboration and responds to user’s contributions based on their last contribution plus their emotional feedback in response to the previous contribution performed by Arny. The turn-taking pattern makes it possible to only focus on expectations from Arny and emotions triggered by Arny in the user and ignore the user’s intrinsic emotions elicited based on how they perceive their own contributions. The turn-taking pattern allows Arny to perceive the user emotional response to each specific contribution made by the AI agent and use it in its next cycle of decision making without confusing them with participant’s emotions about their own contributions.

The current version of Arny’s interaction model, as illustrated in Figure 2, follows a turn taking strategy and selects the next contribution from the AI agent based on three inputs: 1) a memory of the collaborator’s valence in the previous contribution cycle that allows Arny to perceive the user’s value judgment of that last contribution, 2) the user’s engagement level in the beginning of the current cycle that allows Arny to predict user’s expectation from Arny in the current cycle, and 3) the collaborator’s latest contribution to the task in the current cycle. In other words, the first input allows Arny to evaluate how satisfied the user is with the previous interaction pattern the agent followed and if the same type of contribution should be followed or the pattern has to change. The second input on the other hand, allows Arny to know if the user expects Arny’s assistance for discovery of new ideas or if they can continue without a major contribution from Arny. This input was added to the Arny’s latest iteration after participants testing the previous interaction model referred to their expectations of help from Arny as a trigger for parts of their emotions. Finally, the user’s drawing contribution is the third input, to shape the functional sensemaking around the artifact.

Emotion interpretation and decision making suggested by Arny’s interpretation of these inputs by collaboration rules are formed based on our early iteration pilot tests. Arny’s collaboration rules have a strong reliance on engagement.
both for interpretation of the valence value and understanding of the participant expectation after they pass their own turn. Based on the retrospective protocol in our pilot studies the main two triggers for negative emotions in a user are the AI causing an unpleasant distraction from the user’s idea or the AI not meeting the user’s expectation for help in idea- tion. Arny distinguishes these two triggers based on the user’s flow of user engagement and responds to them differently. In these rules, negative value of emotion is considered aligned with any form of emotion that includes a negative value judgment of the process such as confusion, frustration, disappointment, and annoyance. Positive valence value, on the other hand, is in line with emotions that represent a positive value judgment of the collaborator actions including interest, satisfaction, excitement, convinced, and expectant. Arny’s collaboration rules are shown in Table 1. In conditions where the user’s emotion is positive and the engagement level does not reflect a desire for creative ideations from Arny, a converging action is more likely to maintain the positive feeling about the collaboration. In the case when the positive valence is followed by low engagement from the participant, there is a risk of boredom or distraction from the task and so a diverging action is presented by Arny.

<table>
<thead>
<tr>
<th>Cycle</th>
<th>Participant’s Drawing in Cycle N</th>
<th>Participant’s Engagement Level in the beginning of Cycle N</th>
<th>Participant’s Valence Value in Cycle N-1</th>
<th>Arny’s Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycle 1</td>
<td>Participant’s Drawing in Cycle 1</td>
<td>Not referred to for the first cycle</td>
<td>No Previous Emotion to Reference</td>
<td>Converge to Participant Drawing in Cycle 1</td>
</tr>
<tr>
<td>Cycle N</td>
<td>Participant’s Drawing in Cycle N</td>
<td>Medium to High</td>
<td>Positive</td>
<td>Converge to Participant Drawing in Cycle N</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Low</td>
<td>Positive</td>
<td>Diverge from Participant Drawing in Cycle N</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Neutral</td>
<td>Converge to Participant Drawing in Cycle N</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Negative</td>
<td>Pass with No Drawing Action</td>
</tr>
</tbody>
</table>

Table 1: Emotion interpretation and decision making in Arny’s early model

Discussion

Emotion interpretation and decision making in AI-based co-creativity is possible due to recent developments in technology that detects emotion from facial expressions using a web cam. Even though this technology is still being developed and improved, it presents a new direction for computational creativity. In designing the interaction model for Arny, we were influenced by our observations of human to human co-creativity, where the ability to detect emotion is often not noticed or articulated by the collaborators. As a result, our interaction model is based on our assumptions of when a specific emotion should trigger the AI to converge or diverge from the human collaborator’s most recent contribution to the creative task. While we are iterating on the design of Arny based on the reflections by the users of our early versions of Arny, we are also aware of the challenges that simple collaboration rules present, and the difficulty in recognizing how to select a converging or diverging contribution. However, we feel that an interaction model that includes emotion recognition is a first step in developing more emotionally aware AI-based co-creative systems.

References

**Emojinating Co-Creativity:**
Integrating Self-Evaluation and Context-Adaptation

João M. Cunha, Pedro Martins, Nuno Lourenço and Penousal Machado

CISUC, Department of Informatics Engineering
University of Coimbra

{jmacunha,pjmm,naml,machado}@dei.uc.pt

### Abstract

Co-creative systems are useful in fostering creativity, often leading to unexpected results. Despite this, the relation between user and system is complex. The level of autonomy given to the system directly influences its potential for creative behaviour and degree of contribution to the cooperation with the user. In this paper, we present our efforts to instil a more creative behaviour to an existing visual blending system – Emojinating. In order to do so, we integrate two functionalities: self-evaluation and context-adaptation.

### Introduction

Upon the development of creative systems for the visual domain, one of the biggest issues concerns the dependency on human perception – there is no optimal solution as quality depends on the preferences of the user. One approach that has been seen as suitable for such open-ended problems is Interactive Evolutionary Computation (IEC) (Parmee, Abraham, and Machwe, 2008), in which the evolutionary process relies on human evaluation.

On the other hand, when using an Evolutionary Algorithm (EA), researchers are faced with many challenges concerning the configuration and parameterisation, e.g. which operators should be used. One possible way to tackle this challenge consists in using a trial-and-error approach, where the practitioner experiments with several configurations and then select one that achieves reasonably good results. The need to remove this trial-and-error process led to the emergence of adaptive and self-adaptive algorithms. One of the first EAs to introduce this concept was Evolutionary Strategies (ES). In concrete, ES used a mechanism that adapted the rate with which operators were applied. Over the years many mechanisms have been proposed to adapt all other components of the EA (Kramer, 2010).

The combination of user interaction and system self-adaptation provides an adequate setup for a co-creative relation between human and computer. Different types of collaboration are accepted in such co-creative systems (e.g. partnership or assistantship), which vary in terms of complexity of the relation between human and computer, but also on the level of autonomy given to each of them. Instilling a self-adaptive behaviour to the system may increase its contribution in the co-creative relationship with the user.

In this paper, we build upon a system (Emojinating) that uses visual blending of emoji to produce visual representations of concepts. Cunha et al. (2019) presented an interactive evolutionary version of Emojinating, which allowed the user to interact with the system and evolve solutions that fit his/her preferences. However, the system could be said to be more close to a creativity support tool than to a co-creative system, in the way that it mostly responded to user requests. Our goal is to focus on the creative features of the system, leading to an improvement in the co-creative relation.

Our main contributions are: (i) the addition of an automatic evaluator to the evolutionary process, capable of self-evaluating the solutions and adapting to user preferences, and (ii) the introduction of context-adaptation methods. We describe the development of these methods and provide a general analysis of the results obtained.

**Background** The Emojinating system has three main components: (i) the *Concept Extender* (CE), which uses ConceptNet (Speer and Havasi, 2012) to retrieve related concepts to a given one; (ii) the *Emoji Searcher* (ES), which uses EmojiNet (Wijeratne et al., 2017) to retrieve existing emoji that are semantically related to a given word; and (iii) the *Emoji Blender* (EB), which takes two input emoji (Tweemoji 2.3) and produces visual blends.

In general, the system receives a concept from the user that is mapped to two emoji (e.g. emoji A and emoji B), which are then combined through a process of visual blending – emoji B is considered as the base for the blending and emoji A as the replacement. Three different types of operation can be used in the visual blending process: juxtaposition (JUX) – two emoji are put side by side or one over the other (e.g. *peace accord* in Fig 1); replacement (REP) – emoji A replaces part of emoji B (e.g. *health risk* in Fig 1); fusion (FUS) – two emoji are merged together by exchanging parts.

---

(e.g. airline bureaucracy). Three types of operation are used: part exchange (part of emoji B is replaced by part of emoji A), part addition (part of emoji A is added) and part removal (part of emoji B is removed).

The initial version of the system (Cunha, Martins, and Machado, 2018) had a deterministic nature. Cunha et al. (2019) used an interactive evolutionary approach to improve the search space exploration, combining a standard Evolutionary Algorithm with a method inspired by Estimation of Distribution Algorithms (EDA). On a macro level, the system uses the EDA-inspired method to direct the search to areas that match the user preference by stimulating weights assigned to emoji and concepts, based on user fitness assignment. On a micro level, the system is able to focus the evolution on certain individuals by allowing the user to select individuals to be mutated. The user is able to conduct two different actions: select individuals, which increases their fitness, affecting the weight system and producing offspring through mutation; or store them in the archive to avoid losing them in the evolutionary process. For more detail, we refer the reader to Cunha et al. (2019).

**Approach**

Over the last few years, several works have also addressed the evaluation of creativity in co-creative approaches (Jordanous, 2017; Karimi et al., 2018). Two aspects are often considered as requirements in a co-creative system: synchronous collaboration (Davis et al., 2015) and a proactive contribution from both the user and the AI agent (Yannakakis, Liapis, and Alexopoulos, 2014). This means that both agents engage in the interaction and actively contribute to the creative task. Moreover, not only is it required that each agent expresses its own creative ideas but also that it perceives other agents’ contributions (Karimi et al., 2018).

The version of the system proposed by Cunha et al. (2019), despite being able to evolve solutions that match the user taste, has a somehow passive behaviour, as the actions of the system are mostly directly triggered by the user. In this paper, our goal is to enhance the creative behaviour of the system, increasing its autonomy and improving the cooperative character of its interaction with the user. In this way, we intend to instil into the system the capability of adequately responding to user actions, thus improving the co-creative relation.

Briefly describing, the previous version of the system could be said to have two agents: an evaluator (user) and solution generator (system). The version described in this paper introduces a second evaluator (system) that is able to select solutions based on its own idea of quality and storing them in its archive. In addition, we improved the solution generator, increasing its ability to adapt to the context. In this section, we describe the changes conducted.

**Representation**

In the previous versions of the system, only juxtaposition and replacement blend types were implemented. In this version, we improved the blending process by changing the representation used in order to include fusion.

**Variation Operators**

The system presents the user with a population of 20 individuals (blends) and in each generation the user selects the ones to go through a process of producing offspring. Two different operators exist: crossover and mutation. The produced offspring individuals from both operators are added to an offspring pool. A maximum percentage of the new population (50%) is reserved for the offspring, which are randomly selected from the pool. The remaining percentage corresponds to individuals generated from scratch.

**Crossover Operator** A crossover occurs when the user selects at least two individuals. Initially, the system only conducted crossover with individuals that shared at least one emoji. Afterwards, we realised this approach severely reduced the possible offspring. As such, we decided that blends with no shared emoji could also be combined.

In order to conduct the crossover, groups of two are randomly made with the emoji selected by the user. Two types of crossover can occur: if the number of exchange genes (second chromosome) is equal to one, one of the emojis of each parent individual is exchanged with the other individual; if the number of exchange genes in both emojis is above one, it conducts a gene crossover. A gene crossover consists in exchanging genes between individuals, using a one-point crossover. The resulting offspring individuals are added to the offspring pool.

**Mutation Operators** In the previous version of the system (Cunha et al., 2019) only three types of mutation existed (replacement emoji, replaced layer and blend type mutation). With the implementation of the new representation, the types of mutation increased to ones presented in Table 1.

**Adaptation**

Two types of adaptation can be said to exist: to the user and to situations within the system (context). The former has been addressed by Cunha et al. (2019). One of our goals is to focus on the latter, allowing the system to adapt to the
population at the moment, as different stages in the run may require different behaviour from the system. Two different means of context-adaptation were implemented: adaptive blending process (individual generation) and adaptive variation operators (mutation).

The adaptive blending process consists in changing the likelihood of a given type of blend occurring, according to the state of the population. This is used in the generation of new individuals from scratch. The types of blend have different variation potential (juxtaposition has the lowest potential and fusion has the highest). Due to this, our approach is that blend types with higher variation potential should occur more frequently when there are fewer different emoji used in the blends of the population. As such, we assign the probability of each blend type based on the number of different emoji (N_E) for N_E ≥ 20 (higher), JUX = 10% and FUS = 20%; for N_E ≤ 8 (lower), JUX = 2% and FUS = 50%. For 8 < N_E < 20 the following equation is used to calculate probabilities:

\[
\text{LOWER}_{\text{VAL}} + (\text{UPPER}_{\text{VAL}} - \text{LOWER}_{\text{VAL}}) \times \frac{(N_E - 8)}{12}
\]

where LOWER_{VAL} and UPPER_{VAL} are the probability values of the blend type used in the lowest and highest bounds of N_E (e.g. in JUX 2% and 10%, respectively). The value for replacement is always REP = 100 – JUX – FUS.

Regarding mutation adaptation, our initial approach was similar to the adaptive blending process: we tried to assign the same value to each operator and change it according to the state of the population. Later we concluded that due to the characteristics of the problem, this approach would not lead to good results – each type of blend has its own particularities and, therefore, has different mutation requirements. For example, in juxtaposition mutating the replaced emoji is simple as the whole emoji is used, whereas in fusion it is more complex as the layer-based exchanges are relative to the array of layers of each emoji, which varies in number of elements – mutating the replaced emoji in fusion would result in something entirely different. As such, mutation adaptation consists in changing the occurrence probability of each mutation operator according to the blend type of the individual being mutated. We established values for each mutation, depending on the type of blend of the parent (see Table 1). The emoji mutations are independent of the rest. If juxtaposition occurs, none of the rest occurs. If no juxtaposition occurs, any of the other mutation types can occur.

### Self-evaluation and selection

In order to give some autonomy to the system, we decided to bring another agent to the evolutionary process. This agent is an automatic evaluator that has two possible actions: evaluate individuals according to its preferences and store individuals in its own archive. The user can see the archive and is able to retrieve individuals from it but only the automatic evaluator is able to add individuals.

<table>
<thead>
<tr>
<th>mutation type</th>
<th>JUX</th>
<th>REP</th>
<th>FUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>replacement emoji is changed</td>
<td>40</td>
<td>30</td>
<td>5</td>
</tr>
<tr>
<td>base emoji is changed</td>
<td>40</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>blend type changes to juxtaposition*</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>replaced layer is changed</td>
<td>20</td>
<td>35</td>
<td>25</td>
</tr>
<tr>
<td>RP changes from whole to a layer</td>
<td>10</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>RP is changed by selecting a new layer</td>
<td>0</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>RP changes from a layer to whole</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
</tbody>
</table>

### Quality Assessment: Criteria

Defining criteria for quality assessment of the blend is not an easy task. First because quality is dependent on visual attributes but also on conceptual aspects (e.g. does the user perceive the concept). Moreover, as they depend on user understanding and perception it makes this an open-ended evolution problem. In this paper, we chose to focus on the first type of criteria (visual attributes). We considered two aspects that are related to the quality of an icon: complexity (the simpler the better) and legibility (should be perceivable in smaller sizes). Also, given that we are conducting visual blending, we need to also consider the degree of change in comparison to the parents. With this in mind, we defined the following criteria:

1. overall complexity: \( \frac{1}{\text{#BLEND}_{\text{LAYERS}}} \);
2. area exchanged: \( \sum_{i=1}^{\text{#LE}(b)} a(l_i) \);
3. relation between added area and added layers: \( \sum_{i=1}^{\text{#LA}(b)} a(l_i) \)
   \( \frac{\text{#ADDED}_{\text{LAYERS}}}{\text{#ADDED}_{\text{LAYERS}} + \text{#REMOVED}_{\text{LAYERS}}} \).
4. difference in complexity: \( \frac{\text{#LE}}{\text{#ADDED}_{\text{LAYERS}} + \text{#REMOVED}_{\text{LAYERS}}} \).

**Quality Assessment: Fitness Calculation** The goal of the automatic evaluator is to be able to assess solutions based on its own idea of quality. In this sense, there are two options: having an evaluator that tries to get similar solutions to the user, in order to present good alternative solutions; or get solutions that are distinct from what the user is selecting. In this paper, we chose to focus on the first approach.

The system’s idea of good solutions is therefore dependent on user choices. This is achieved by making the system analyse the blends in the user archive – which are assumed as being good – and afterwards change its idea of a good solution to match these user-selected blends. In the beginning, the system starts with default values (all equal to 1). As soon as the user stores individuals in the archive, the system evaluates them and changes its fitness goal, based on their characteristics. To obtain the goal, the system calculates the average of each criterion for the individuals stored in the user archive, which results in an average blend profile. This profile is then set as the new goal and used for selecting individuals that the system finds interesting. As such, the system goal changes over time, according to user preferences. In order to calculate the fitness of an individual, the system uses a Euclidean distance between the individual and
the average profile, which assesses how far away the system is from the goal and is updated at the end of each generation.

**Individual selection** As already mentioned, the automatic evaluator has its own archive. At the end of each generation, and after calculating the new goal, the system performs an analysis of the population to check for good individuals. The evaluator’s archive capacity was set to 5 to avoid storing too many individuals. Also to avoid collecting too many individuals, the evaluator only stores one blend per emoji combination. This way, the system tries to improve the fitness of individuals for each emoji combination.

In each generation, the evaluator selects the best blend in the population, checks whether it already has a blend for the emoji combination and proceeds as follows: (1) if there is already a blend for the emoji combination and the fitness of the stored one is lower than the current population best, it replaces the individual in the archive; (2) if it does not have any blend for the emoji combination and the archive has free space, the evaluator stores the blend; (3) if it does not have any blend for the emoji combination but there is no space in the archive, the evaluator checks if the blend to store has higher fitness than the worst individual in its archive and, if so, replaces it. In each generation, the evaluator discards individuals when the difference of their fitness to the best individual in the population is >2 (empirically obtained), which often happens when the fitness goal drastically changes.

**Discussion and Future Work**

As this paper presents work in progress, the results of the system are, at this point, qualitatively evaluated and discussed by the authors.

In general, the system is able to learn from the user behaviour, which is observed in the storing of similar blends in its archive (e.g. if the user selects blends with a large exchanged area, the system tends to replace the blends in its archive to match the user preference). Moreover, the system archive is useful to highlight blends that the user may have missed as the system only selects blends that were previously shown to the user. However, there are some improvements that need to be made, e.g. the evaluator should avoid blends that are exactly the same to any stored by the user. Regarding the evaluator behaviour, we explored a strategy of searching for blends similar to the user but another possible approach is to try to go in directions that are different from the user’s, in order to increase the variety of results. Moreover, a possible future direction is to allow the automatic evaluator to select blends to be reproduced, generating offspring from its own stored individuals.

The values used in this paper were empirically obtained through experimentation and adjustments. However, due to the high number of parameters we consider that further tuning is required. One example is the probability of fusion, which depends on the number of different emoji in the population. Its probability of occurrence was set to a high value (50%) for low emoji number, as it is the type of blend that leads to the highest variety of results – theoretically, this would be suitable in situations in which few emoji exist. However, this does not work when put in practice as it makes it harder to identify both parent emoji (e.g. airline bureaucracy in Fig. 1), which worsens the user perception from the first generation. A possible solution may be to also consider the number of the current generation.

Concerning fitness, there are also some issues that need to be addressed. First, using an average of individuals’ properties as a goal does not work well for every case – an individual located between two good individuals is not always a good individual. Even more, as we are dealing with pictographs, in which the perception of the concept is more important than some of the considered visual features (e.g. area changed). Further studies are required to better validate the implemented approach. In future work, we will conduct testing with users to understand how the co-creative functionalities help in obtaining better results. A video of the system being used can be seen at https://rebrand.ly/iccc20short.

**Acknowledgements** This work is partially funded by the FCT - Foundation for Science and Technology, I.P., within the scope of the project C1SUC - UID/CEC/00326/2020 and by European Social Fund, through the Regional Operational Program Centro 2020, and under the grant SFRH/BD/12905/2016.

**References**


Jordanous, A. 2017. Co-creativity and perceptions of computational agents in co-creativity. In Proc. of the Eighth Int. Conf. on Computational Creativity, Atlanta, US. ACC.


2. Design and CC
Exploring the flexibility of a design tool through different artificial agents

Gerard Serra, David Miralles, Maria Casals and Míriam López
Grup de recerca en Tecnologies Media
La Salle-Universitat Ramon Llull
Quatre Camins, 30 08022
Barcelona, Catalonia, Spain
gerard.serra@salle.url.edu

Abstract
Many machine learning approaches are focused on defining artificial agents able to find solutions to a certain problem given fixed design tools or parameters to optimize. In order to do that, creators must have certain knowledge of the solution space to define design parameters that ensure enough exploration allowing agent to find its best configuration. However, this approach may limit artificial agents since they are restricted by their initial conditions of a certain design problem. In addition, specific initial conditions also limits them to scale across multiple challenges.

In this paper, we explore how the definition of more general design tools can allow artificial agents to better explore the solution space and generalize through multiple design problems. To do that, we compare design artifacts produced by an artificial agent that learns to construct 2D shapes with a fixed number of pieces to another artificial agent that also learns to add or remove pieces from its design proposal. We demonstrate how by allowing more freedom in design, an artificial system is able to produce more novel artifacts with higher performances in multiple scenarios.

Introduction
Design can be described as a process of co-evolution of both problem and solution spaces (Maher and Poon 1996; Dorst and Cross 2001; Howard, Culley, and Dekoninck 2008). In this iterative process that involves generating and evaluating solutions, knowledge is acquired, augmenting designers’ capabilities to generate creative designs based on previous experience (Gero and Kannengiesser 2004; Boden 2004). To support their creative role (Dorst and Cross 2001), many computational design tools have been defined (Philippa and Michael ) ranging from early design exploration to more advanced design phases such as refinement and optimization. In this work, we focus on the creation of computational design tools that can allow an artificial agent to explore the design space for a given problem. We inherit from Wiggins (Wiggins 2006) formalization about Boden’s creativity concepts (Boden 2004) considering the exploration of possible design proposals, named artifacts, mainly as a search in a conceptual space. This way, a the set of rules and actions to generate artifacts must be defined, playing a crucial role in the solution space exploration. However, this definition is often strictly related to an specific problem space limiting system’s possibilities to generalize to multiple problems. In our approach, problem space will be defined in (Serra and Miralles 2019) environment that allows us to test 2D shapes as design proposals on multiple physically based design problems. We decided to use this environment since we can define different types of problems such as collecting elements, moving through or protecting areas which has already been resolved by humans. We are very interested on the possible emergence of human solutions for this problems but specially new solutions that may inspire future proposals. In addition, physically based environments allow us to simulate how different proposals may behave in a future real scenario. This is specially relevant to understand why some proposals perform better and which constrains may appear during the exploration of the solution space. To generate these proposals, we explore how the definition of different design tools directly affects on system’s generative and learning capabilities. We are specially interested in how the same computational tool can be applied in multiple problems. In order to define these tools, we consider that flexibility is one of the key aspects to allow computational systems to explore problem space and re-adapting from possible non-favorable initial conditions while generalizing better across different scenarios. As shown in (Ha 2019), by allowing an agent to learn optimal physical configurations for a given task it improves its performance and it facilitates its policy learning. To do that, in our experiments, we compare different population-based search algorithms with two different constructive methods. The first one, based on learning to optimize a shape with an already defined number of pieces. This approach can benefit the algorithm to find solutions, but it requires that creators know the problem space, since a possible number of pieces must be proposed for the solution. In contrast, the second method consists in allowing the agent to freely modify its shape by adding or removing pieces. We evaluated each method capabilities to generate creative designs by comparing their artifacts produced considering both their performance and novelty (Ritchie 2007; Maher and Fisher 2012). Although the first constructive method is more efficient in finding possible solutions, the second method can even provide more novel valid proposals in multiple scenarios. By the combination of modular blocks within multiple environment and without any previ-
ous knowledge, our system has been able to generate design proposals from scratch that resemble human proposals. These results show the importance of defining tools that can perform more actions to explore the solution space rather than focusing solely on the complexity of the algorithm.

Related work

Our work is based on previous research on evolutionary computing (Eiben, Smith, and others 2003) that has demonstrated its capabilities to solve complex challenges in multiple environments (Lehman et al. 2018). These techniques have been widely used in robotics ranging from optimizing already defined morphologies (Joachimczak, Suzuki, and Arita 2015; Nygaard et al. 2018) to even generating them from scratch (Sims 1994; Lipson and Pollack 2000). We highlight the work on Evolutionary Design from (Bentley 1999), showing how evolutionary strategies can also be applied to producing novel and functional designs. To do that, a formalism must be defined in order to explore actions on certain solution space (Wiggins 2006). We draw inspiration from Shape Grammars firstly introduced by (Stiny 1980). A Shape Grammar (SG) consists of a computational formalism that allows automatic shape generation by providing a finite set of shapes and rules that will be applied to these shapes. By continually applying these rules, original shapes are transformed and complex structures can emerge from this process presenting similarities to creative design process theories (Maher and Poon 1996; Dorst and Cross 2001; Gero and Kannengiesser 2004). SG have been often combined with Evolutionary Algorithms (Duarte 2005; O’Neill et al. 2010; Lee and Tang 2009). This combination has also been referred as a Grammatical Evolution (GE) (O’Neill and Ryan 2001; Dempsey, O’Neill, and Brabazon 2009). The main advantage of combining a Shape Grammar with an Evolutionary Algorithm is its exploratory capabilities of the solution space. Many of this projects focus on the generative power of shape grammars rather than their possibilities as a design language (Knight 2000) that can be shared between humans and artificial agents. In addition to that, previous research lines are focused on solving an specific problem rather than aiming a more general knowledge acquisition (O’Neill et al. 2010; Lee and Tang 2009). Other studies (Ha 2019) have also explored how by allowing agents to also optimize their initial design conditions they can perform better given a certain problem. Specially the work of (Pathak et al. 2019) demonstrates how a better generalization can be achieved by the usage of modular elements to construct. In our approach we want to define a computational tool that can be used for both humans and artificial agents to solve multiple problems. In contrast to other previous work, we are interested both in the performance and the novelty of artifacts produced with our tool and its capabilities to be used in multiple problems. In addition, we want to show how, by being less restrictive and allowing more actions, our tool has a direct impact on the emergence of design by providing a wide range of creative solutions without losing performance.

Methodology

We have performed our experiments in Coevo environment (Serra and Miralles 2019) that allow us to test a collection of physically based scenarios with specific design problems. This environment has been inspired by (Brockman et al. 2016) with the focus on 2D design creation allowing both humans and artificial agents to generate proposals. Each scenario has a fixed simulation time and conditions to evaluate design proposals. A total of five scenarios have been proposed as a benchmark for our comparative study (Figure 1).

- **E0 - Collect balls.** Each proposal is evaluated by the number of falling balls collected. We have two variants based on design proposal position: left side (E0.1) or at the middle of the scenario (E0.2).

- **E1 - Move along an inclined plane.** Each proposal is evaluated based on the total distance moved within the simulation steps that the experiments lasts.

- **E2 - Move through a different medium.** Each proposal is evaluated on total distance moved from an initial free fall position and experimenting a drag force $\vec{F}_s$ when entering the different medium (Equation 1).

$$\vec{F}_s = -sv^2A \hat{v}$$ (1)

where $s$ refers to the medium properties (density and drag coefficient), $v$ refers to the speed of the proposed shape, $A$ the frontal surface of the shape that is pushing through the new medium and $\hat{v}$ the velocity unit vector. The area is computed by extracting the cross section of the proposal’s shape. Then, the greater the area, the more difficult to reach the bottom.

- **E3 - Protect area.** Each proposal is evaluated by counting the number of randomly generated balls that hit the highlighted orange area (Figure 1).

These scenarios provide diversity within the solutions that may emerge on the solution space exploration. While E0 and E3, require larger number of blocks to be solved, E1 and E2 perform better with less number of blocks. Based on this knowledge, we can define constructive rules and initial configurations that allow the system optimally solving the scenario. However, this definition may not be optimal for other scenarios or even limit system capabilities to solve other specific scenario. So, we want to explore how to define a system that can perform better globally while allowing a proper exploration of the problem space. This aspect will be discussed in the following sections.

Design tool

As we mentioned earlier, to generate design proposals we use a language inspired by shape grammar formalism (Stiny 1980). This grammar is defined by the following elements:

- **Initial shape:** single block with fixed dimensions.

- **Shape:** finite set of proposals defined by rules

- **Rule:** transformations applied to shapes (Figure 2).
Figure 1: Scenarios. From left to right: collect falling balls, move along inclined plane, move through different medium and protect target.

Figure 2: This shape grammar consists of three basic rules: add block (A); remove block (B); block rotation (C).

These constructive rules can allow emergence of complex shapes by concatenating simple blocks. In order to allow more freedom in design, block overlapping has been permitted.

Once the language has been defined, an internal representation of these operations and rules has been created in order to allow an artificial agent to learn how to construct a proposal (Figure 3). In our approach, each shape is described by one integer which corresponds to the length of the unit block, followed by an open-ended stream of angles in radians which describe all the shape, as is showed in the equation (2):

$$\text{design proposal} = [\text{length}, \phi_1, \phi_2, \ldots, \phi_N]$$  \hspace{1cm} (2)

where $N$ is the number of blocks that make up a shape.

Figure 3: Internal representation of three different shapes generated using this shape grammar.

In our experiments, we maintained block length to 30 units to remain consistent across all the scenarios and artificial agents.

Artificial Agents Definition

For this study, we created three different artificial agents that learn to generate design proposals using previously described tools. All these AI agents are based on evolutionary techniques that learn to optimize its shape to fit the problem of each scenario. Then, we summarize each agent used:

- **Fixed Genetic algorithm**: this agent is based on a simple genetic algorithm (Mitchell 1998) that selects best candidates using roulette-wheel selection via stochastic acceptance (Lipowski and Lipowska 2012). Crossover is performed by combining selected candidates representation and we also add a mutation value that randomly changes angles ($0, 2\pi$) to add noise when defining a new population.

- **Fixed CMA-ES**: this agent is based on Covariance-Matrix Adaptation Evolution Strategy (Hansen, Müller, and Koumoutsakos 2003) adapted to optimize a shape with a certain number of blocks. Initial population is randomly generated. Then, each new population is generated within time from multiple distribution of mean and covariances (one for each block) based on previous generation performance. Note that the number of distributions depends on the initial number of blocks defined for that certain experiment.

- **Variable Genetic algorithm**: similar to the first agent, this approach is also based on a genetic algorithm. Its main difference is that a mutation value for adding and removing pieces has been also added. This allows the agent to optimize also the number of pieces required and explore possible valid morphologies for each scenario.

We have chosen to define our agents based on these evolutionary techniques as being ones of the most simple and popular amongst researchers in the field. (Salimans et al. 2017; Prabhu et al. 2018). In addition, population-based search techniques make possible to explore many areas in these spaces at once (Miikkulainen 2019) so we have considered ideal for our experiments. All experiments are initialized with a fixed number of blocks and only the third one is able to add and remove blocks. This decision allows us to evaluate how an agent with more design capabilities performs in comparison with the other ones.

Experiments

Here we enumerate all the experiments performed with each artificial agent and scenarios described in previous section.

As seen in Figure 4, each scenario and algorithm has been initialized with three different number of blocks (6, 12 and 24, respectively). Each combination has been simulated for 200 generations with a population of 100 members.
each one initialized randomly at the beginning of the experiment. We decided to define this initial conditions to compare how the different agents behave in possible optimal or bad initial configurations. Since we are specially interested on the global performance and novelty of each agent we consider that initializing the agent with three different number of blocks gives us a general idea on how the agent is able to adapt and provide different solutions within a limited number of iterations (200 generations). Finally, we repeated each experiment 10 times to have enough data to extract design patterns. This makes a total of 450 experiments to be analyzed.

![Figure 4](image)

Figure 4: A total of 45 combinations can be performed considering given variables: scenario, agents and number of blocks

All design proposals are placed in the same corresponding initial position and evaluated individually using each scenario specific fitness function.

Results

In this section, we present the results based on the design proposals generated by each agent. Our goal is to evaluate agents capabilities to produce creative designs. We have considered (Ritchie 2007) approach for evaluating individual creativity by its produced artifacts rather than from the process used. To do that, we evaluate each individual artifacts based on (Maher and Fisher 2012) proposal that considers three parts for evaluation:

- **Value**: performance measure of the design.
- **Novelty**: similarity from the rest of the proposals.
- **Surprise**: how an artifact can exceed the value and novelty expectations of the already defined patterns found in the solution space.

Value

Since we have captured and analyzed all the designs produced we measure the value by computing the fitness obtained by the best member of each generation from each experiment. This parameter also gives us an idea of each agent’s performance within generations. As we can see in Figure 5, there is a common behavior between configurations with fixed number of initial blocks reaching high fitness in most scenarios when their number of blocks is optimal for that scenario. In contrast, performs worse when this initial number is not optimal. For example, In Scenarios 0.1 and 0.2, only the configurations that start with 24 blocks are able to reach higher fitness. This behavior is also seen in Scenario 3, in which configurations with higher amount of blocks perform better. Opposite to that, in Scenario 2, configuration with lesser number of blocks perform better reaching maximum fitness faster. In this Scenario 2, the agent based on GA-24-fixed is the only place where this agent does not find a solution. We also observe that both fixed agents are also able to reach higher fitness within generations with the exception of the fixed ones that started with only 6 blocks.

<table>
<thead>
<tr>
<th>Fixed G.A</th>
<th>CMA-ES</th>
<th>Variable G.A</th>
</tr>
</thead>
<tbody>
<tr>
<td>E0.1</td>
<td>0.24</td>
<td>1</td>
</tr>
<tr>
<td>E0.2</td>
<td>0.3</td>
<td>0.71</td>
</tr>
<tr>
<td>E1</td>
<td>0.43</td>
<td>0.97</td>
</tr>
<tr>
<td>E2</td>
<td>0.65</td>
<td>1</td>
</tr>
<tr>
<td>E3</td>
<td>0.12</td>
<td>0.88</td>
</tr>
</tbody>
</table>

Table 1: Comparison between worst fitness and best fitness obtained by each agent configuration. As shown, agent with variable number of blocks performances are more similar.

In contrast to that, as also shown in Table 1, the agent with a variable number of blocks is able to perform better, no matter the number of blocks it is initialized. Similar to previous agents, proposals generated by this third agent are able to reach higher fitness in all scenarios expect from Scenario 0.2, also the worst scenario for the other agents. Using its constructive capabilities is able to optimize the number of blocks needed to solve the scenario. One exception to this behavior is in Scenario 1 with the with the agent starting form the lowest number of blocks (6), that the third agent has not been able to reach higher fitness.

Novelty and surprise

In terms of agent novelty, we have decided to evaluate each group of generated design proposals based on how similar are from each other. This approach is based on (Maher and Fisher 2012) which proposes evaluating similarity using distance of potentially creative designs and later on clustering them based on that. Since each agent proposes a large number of designs, we have only considered the ones a threshold performance higher than 0.9. In addition, to further reduce the number of proposals, we randomly pick only 15 proposals of each roll-out for agent. This ensures having enough representatives of each agent agent while maintaining a small data set for our similarity comparison. Then we have multiple data sets of valuable design proposals generated by each artificial agent. These two decisions ensure that selected design proposals are valuable to the given problem, while we can also evaluate how different they are across agents.

Then, we must define an efficient comparison method for determining which data set contains more novel designs. To do that, we decided to generate an image containing each proposal. To ensure enough resolution, we centrally place each proposal in a 300x300 image and then we reduce their dimensionality into 2 components using Principal Compo-
Figure 5: Learning process from each scenario and agent configuration considering 10 random rollouts

Figure 6: Random artifacts selected from third agent (G.A Variable) proposals. As shown different proposals can emerge from simple parts in each scenario.

Component Analysis (Wold, Esbensen, and Geladi 1987). This reduction helps us to visualize similar proposals closer in a 2D space allowing us to navigate between them and understand better similarity relations. We decided to use this approach to standardize all the design proposals within a single measurement since each agent may provide solutions with different number of pieces. We computed this value using (Pedregosa et al. 2011) tools. Then, for each scenario, we have placed each proposal on 2D dimensional space based on these two PCA components and clustered them.

For clustering, we propose Mean Shift algorithm (MS) (Comaniciu and Meer 2002). We decided to use the MS algorithm because it does not predetermine the number of clusters. We are interested in the emerge of clusters from our current distribution of proposals. Then from each cluster we randomly selected multiple representative proposals to compare them visually. As an example of this selection Figure 6 provides a visual overview on the divergence of solutions present in each scenario.

Regarding novelty between agents, in general, there is not a significant differentiation between the novelty produced by agents with fixed initial number of blocks than the others. All three agents are able to produce similar design proposals considering their number of blocks. However, as observed in Figure 7, the number of blocks strongly conditions the shape of generated proposals. As an example, proposals
from agent that CMA-ES with 24 blocks resemble a lot each other. In contrast, the agent with variable number of blocks is able to converge to a wide range of solutions with multiple number of blocks. This flexibility in design results in a greater dispersion of the generated artifacts.

Finally in terms of surprise, unexpected results specially emerged on Scenario 1. This scenario was originally designed expecting proposals with a circular shape similar to a wheel. Each design agent has been able to produce not only circular shapes but also a large number of different shapes able to move along given inclined plane.

**Discussion and future work**

In this section, we discuss the results presented before and the findings based on the proposals generated by our artificial agents. We also include current limitations of our approach and plans for future work.

To perform our analysis, we have compared all the designs produced by our three different artificial agents in a total of 5 different scenarios. In general, all three agents have been able to produce valuable artifacts for each scenario. In terms of performance, Scenario 0.2 has been proved to be the most difficult one, directly lowering the performance obtained by the agents. In this particular scenario only the proposals generated by CMA-ES agent with 24 blocks has been able to surpass the value of 0.8 in fitness. In other scenarios, Variable G.A agent has been the unique one that has generated proposals with higher fitness no matter the initial number of blocks. This result evidences how an approach that allow more freedom in designing influences positively in exploration of the design space ending up in a richer number of high valuable generated artifacts. In contrast, both Fixed G.A and CMA-ES agents highly depend on initial parameters having less capabilities to adapt to each scenario. Then, only when initial parameters are beneficial, their performance is better reaching higher fitness faster than the others. One limitation of the current work is related to the initial conditions given to the system in terms of number of blocks and allowed iterations(200). As shown in our results, some of these configurations may limit agent’s capabilities of finding optimal solutions. However, this has not happened in the flexible agent which, despite being affected in the iterations necessary to find optimal solutions, its exploratory capabilities allowed it to find solutions regardless of its initial conditions. This supports our approach that defining flexible constructive methods allows our computational tools to generalise better since we are not embedding scenario specific knowledge that may affect negatively in other situations.

Since the number of valuable proposals has been large in all the scenarios, the definition of metrics and tools to evaluate, compare and clustering them based on similarity has become crucial in our work. We have defined a comparison method inspired by (Maher and Fisher 2012) work on evaluating novelty as a distance between individual proposals. In our approach, we showed how by generating images from each shape and using PCA (Wold, Esbensen, and Geladi 1987) we can efficiently visualize and select proposals for novelty evaluation. Then, regarding novelty, our results evidence how by using these simple design tools a wide diversity of proposals emerge in all the scenarios and agents. This behavior is also stronger in Variable G.A agent since is not influenced by its initial conditions, its solution space exploration is higher. Our results also suggest how population-based algorithms combined with simple design tools inspired by shape grammars can be a powerful combination for iteratively exploring multiple solution spaces.

As we seen in Figure 6, the same tools are able to generate a rich diversity of proposals for each scenario. Then, designers role in this creative environment can be focused on defining problem space and collaborating with artificial agents to propose solutions to propose proper solutions to that problem. Our current environment is limited to only five different scenarios. However, new evaluation techniques can be applied to each of them or even new scenarios can be created and tested using our artificial agents.

An interesting future work would be to explore how problem space definition by designers can influence the novelty of designs generated by artificial agents. It has been shown that the most complex scenario (Move along a plane) is the one that produced a greater emergence of novel design proposals. In natural evolution, the environment plays an important role in diversity, however, more research should be done to determine if this also happens on digital environments.

In future work, we will also investigate how these design tools can also be used by humans and how they can collaborate with different artificial agents to solve together a given challenge. In our work, many artifacts that resemble to human designs have emerged through each artificial agent learning process. However, also unique designs that we have not initially thought about have also emerged. It would be very interesting to explore how human creative capabilities can be augmented by collaborating with agents with no prior knowledge given a design problem.
Conclusions

In this paper we are especially interested in how by providing an artificial agent with more degrees of freedom in its creation tools, it can better adapt to multiple design challenges by offering proposals of greater value and novelty. To do that, we presented a new set of design tools to construct complex proposals by concatenating minimal blocks. The environment provided (Serra and Miralles 2019) together with the tools created, allowed us to define and evaluate problems such as collecting elements, moving or protecting areas that have been already been solved by humans in different ways. Our results suggest that the degrees of freedom given to the tool allowed the system to generate more novel designs with higher performance providing also solutions that are not influenced by initial design considerations based on the expected solution of a given problem.

To show that in our studies, we have defined three population-based different evolutionary agents that have generated design proposals for a total of five different scenarios. Each agent is initialized with a fixed number of blocks that can use to construct and only one has been allowed to change this number during its learning process. By defining the initial number of blocks we are providing some knowledge on the solution space since some environments can be solved optimally depending on this number. However, this knowledge is related to a certain set of solution that the creator may have in mind limiting the system to explore other solution spaces. In addition to that, it cannot be generalized in different scenarios, since this knowledge that can be beneficial in some scenarios is a limitation in others.

As an example, E0 and E3 involve that the solution includes larger number of blocks than scenarios E1 and E2. Then, agents initialized with the optimal number of blocks learn faster than others that may not even reach higher fitness due to their initial definition 5. Then when defining these systems, creators must consider initial configuration as a key aspect in their design. This requires an initial human effort to understand the problem and also an initial limitation since the creators are already embedding their knowledge in the tool they are creating. However, our results suggest that flexible agent does not show this limitation in the given scenarios. In contrast to fixed ones, variable agent is able to reach optimal solution spaces despite the fact of being initialized in a less beneficial solution space or even with a configuration that has no possible solutions to the given problem. As a result of this, our artificial agent has been able to construct valid design proposals across multiple scenarios surpassing the other two agents in terms of performance and novelty. Is also specially relevant that this agent is also able to find novel solutions with high performance compared to fixed agents initialized on optimal spaces. Our results suggest that allowing more degrees of freedom influences the ability to innovate by reconfiguring its morphology, augmenting the space of possibilities and exploring new paths within this space in each scenario. Especially in E1, by continually adding pieces, different new shapes emerge to the wheels, such as spirals or S-shaped morphologies similar to sleds. This phenomenon may be related to the evolutionary path followed by the solutions provided by the variable agent since all the possibilities found by the fixed agent end up in the wheel as an optimal shape.

We hope that our results encourage computational creativity community to continue working on the definition of flexible design tools that allow artificial agents to better adapt to multiple environments.

Acknowledgments

This work has been partially funded with the support of the European Social Fund and the Secretaria d’Universitats i Recerca del Departament d’Economia i Coneixement of the Catalan Government for the pre-doctoral FI grant No. 2019 FI_BI 00171.

References

Bentley, P. J. 1999. Evolutionary design by computers. Morgan Kaufmann.


Philippa, M., and Michael, B. V. An ontology of computational tools for design activities.


Deep Learning as heuristic approach for architectural concept generation

Daniel Bolojan; Emmanouil Vermisso
Florida Atlantic University
School of Architecture
111 East Las Olas Boulevard
Fort Lauderdale, FL 33308
dbolojan@fau.edu; evermiss@fau.edu

Abstract
This work discusses the use of machine learning for extending the boundaries of architectural concepts. The use of deep learning algorithms is exploited for transferring stylistic attributes from one visual (architectural) domain to another (architectural or otherwise). The combination of various semantic features into hybrid results helps locate possibilities for further exploration of one domain. The process is regarded as open-ended, taking advantage of a generative adversarial network’s (cycleGAN) capacity to “imagine” new results which may set the base for new morphological or structural architectural paradigms. We are interested to situate the training results within the context of human and computational creativity, embracing the capacity of automated heuristic methods to produce novel outcomes which can augment human decision making in design and architecture.

Creativity and Artificial Intelligence
The project discussed herewith aims to examine the potential of AI methods for architectural design, acknowledging that the latter warrants a substantial degree of creative thinking, that could be enhanced by machine learning processes. In spite of early interest in machine intelligence and automation in the 19th century, the scientific community only became seriously involved in Artificial Intelligence during the late 1940s and ’50s, prompted by Alan Turing’s work during the Second World War. It seems that the turning point for strong interest from the general public came quite later; most likely, Garry Kasparov’s defeat to IBM’s Deep Blue algorithm in chess in 1997 signified a rupture in the collective perception of the capabilities of human mind and the limitations of machines. (Kasparov, 2017) Although wide-spread claim of a “Computational Creativity” might have been premature, that match set a tone for what was possible in the following years, considering the loss of a human grandmaster had not yet been anticipated. Machine cognition and chess had preoccupied scientists for a long time: in the last chapter of the original 1948 “Cybernetics”, MIT Professor Norbert Wiener contemplated “...whether it is possible to construct a chess-playing machine and whether this sort of ability represents an essential difference between the potentialities of the machine and the mind. Note that we need not raise the question as to whether it is possible to construct a machine which will play an optimum game in the sense of von Neumann. Not even the best human brain approximates to this.” (Wiener, 1965)

Wiener foreshadowed an aspect of AI that would come to the foreground of academic interest much later; a machine that could merely play chess was possible, but to challenge a human player of the highest level, a machine would require some degree of ingenuity - in other words, (computational) creativity. According to experts, Computational Creativity is defined as “...an emerging field of research within AI that focuses on the capacity of machines to both generate and evaluate novel outputs that would, if produced by a human, be considered creative.” (Veale, Cardoso, & Pérez, 2019) Our engagement in defining and assessing creativity within the context of machines can help critique our existing preconceptions about creativity: “…the artifacts that are produced also serve as empirical tests of the adequacy of scientific theories of creativity”.

By extension, understanding the process by which machines learn to produce novel outcomes may help us analyze our own cognitive processes more fundamentally. The notion of “Creativity” is a considerable topic in 20th century academic inquiry, and a number of insights about the behavioral structure that nurtures it have been offered. Mathematician Henri Poincaré proposed a four-stage model for achieving high level thinking: Preparation (conscious thought); Incubation (unconscious thought); Illumination and Verification. (Boden, 2004) These stages are sequential and necessary to stimulate the mind towards creative problem-solving. Margaret Boden has explained that creativity manifests in primarily three different forms, progressing from the most common to the least achievable: Combinatorial, Exploratory and Transformational Creativity. (Boden, 2004) Boden’s work focuses on understanding the limitations of machine “creativity”, described as the context “…in which the computer at least appears to be creative to some degree.” Through this task, Boden believes that “Computational ideas can help us to understand how human creativity is possible.” (Boden, 2004)
Technological Paradigms in Architectural Design Method

The progressive development of architectural design processes which are informed by digital tools has led to the inevitability of incorporating - to some degree - artificial intelligence as the next step in exploring design ideas; starting in the late 1990s, the exponential growth of digital technologies like automated drafting, modeling and fabrication tools (CAD/CAM) changed the architectural arena and rapidly permeated both architectural education and research, as well as practice. It is noteworthy that although the recent themes of state-of-the-art research venues in architectural design computation reflect architects’ intent to be critical and integrative in light of new technologies, tools and processes, the results still largely focus on automated fabrication protocols and material research. While this is important and necessary to advance our practice and probe the industry to innovate, it has inadvertently suppressed a more introspective need to reflect on the design process itself, not only the technical underpinnings of design innovation, but also the cognitive ones, and how these can inform each other.

Symptomatically, technology-driven architectural discourse of the past 10 years describes a tendency to increasingly focus on “synthetic” workflows which use a variety of tools, modes and often, expertise. “Integration” seems a common theme and indeed reflects the desire, and often, necessity on our end to collaborate with other disciplines to fulfill research objectives. The emergence of “parametric modeling” software and its adoption in design (post-2010) has not sufficed to address the numerous and complex problems resulting from the possibilities it offered in the first place. Nevertheless, architects were able to start regarding the design steps as a sequence of parameterization, and the building as a system, with several possible solutions within a large design space, similar to evolutionary models in nature, where only the fittest design survives.

Recently, the nexus of simulation-fabrication has been enhanced by simulated material performance and virtual and augmented reality tools (VR/AR) which increase our participation in various stages of the design workflow (“persistent modeling”). Still, the most promising and disruptive factor so far - albeit not yet exploited - remains the introduction of “Data” as an important contribution for design. This has remained latent, explicitly addressed within limited domains of scholarly inquiry.

The introduction of Artificial Intelligence as a theme for recent specialized architectural research venues reinforces our latent intention to examine the possibility for AI-driven processes to inform design thinking. Among the few examples of AI as a generative tool for imagining architecture is the work of media artist Refik Anadol. Anadol curates “hybrid” environments in existing architectural space using media arts and AI to direct “machine hallucinations”. His work is enabled by large data sets which have been compiled and sorted using machine learning algorithms. (Anadol, 2017)

The emergence of Machine Learning: Connectionism in AI as precursor to Deep Learning

The process we discuss herewith involves training a deep neural network to recognize image features and “imagine” new outcomes. Deep Learning is an area of AI that evolved from Cybernetics and early Connectionism. The transition from the early paradigm of classical Artificial Intelligence (GOFAI), which was inspired by logic and required hard-coded instructions for any task, to the biologically inspired paradigm of neural networks (Connectionism), took longer than originally expected. The earliest work on neural networks began in the 1940s by Warren McCulloch and Walter Pitts, while the first artificial neural network (“Perceptron”) was created by psychologist Frank Rosenblatt in the late 1950s. The use of neural networks started the “connectionist” (biological) paradigm in artificial intelligence. Despite their insight, early A.I. researchers had difficulty predicting the evolution of automated procedures, because these relied on factors like data availability, data accessibility and representational protocols. Furthermore, connectionism was criticized by MIT Professors Marvin Minsky and Seymour Papert in “Perceptrons: An introduction to computational geometry” (1969). Their critique, focusing on “...the lack of adequate basic theories” to support the connectionist approach to machine learning persisted during twenty years; in a revised edition of the book, machine learning expert Léon Bottou pointed the persisting limitations of mathematically describing meaning, even in certain machine learning areas where progress had been made (i.e. computer vision): “Although we can conceivably prove that a program fulfills the heuristic specification, this does not guarantee that the program can perform the task of interest. In the case of computer vision, scientists have devised many ingenious ways to leverage physical and geometrical insights about the nature of images. However, absent a mathematical definition of what makes a cat look like a cat, the logical gap remains. Almost a proof is no proof“. (Minsky & Papert, 2017 (1st ed. 1969))

While heuristics are imperfect for problem-solving, there are certain advantages in this approach. The importance of Machine Learning is understood when we consider the difference between the two AI paradigms; traditional hard-coded tools are problematic when the numbers of possibilities increase exponentially (i.e. Chess vs GO). Machine learning enables an algorithm to not do exhaustive search by brute force, but “learn” to recognize probability through pattern identification.

Margaret Boden discusses the advantages of heuristics when using neural networks. Although the networks’ architectures are known to their designers, their detailed performance-driven adjustments sometimes remain a “black box”. Still, a heuristic pursuit of a problem is useful for pruning the search-tree. (Boden, 2004) According to Professor Michael Dertouzos, late director of the MIT Computer Science Lab, this research strategy does not guarantee an optimal solution but can be useful for beginning to address difficult problems. In his foreword to the book
“What Will Be”, Dertouzos identified a particular tendency in certain cultures, to approach a given problem with a mix of systematic vs. loose thinking. (Dertouzos, 1997) This involves a convoluted way of seeking answers rapidly, then revising errors and refining until it is adequate. This kind of “undisciplined” structure of thinking (in comparison to very systematic work process among i.e. Japanese or Swiss cultures, where most parameters are sought beforehand), can be advantageous for working with Computer Science problems. This is an important observation with regards to building working contexts that can nurture creative thinking and can likely prove beneficial to Architectural Design, because it is an inherently non-linear process.

**Deep Networks Architecture**

The affinity between unsupervised learning in neural networks and human learning process underlines the importance of contextualizing unsupervised learning that could help simulate creative thinking in domains like Design and Architecture. Consultation across disciplines and dissemination of such works among the fields of Computer Science, Neuroscience, and Architecture points to integrative schemas of architectural research. In response to recent demand from the architectural community for generative design inquiry, we tested the use of cycleGAN during a 2-day workshop on machine learning for architectural design. The workshop - titled Gaudi’s Hallucinations - examined a synthetic design process which reviewed aspects of architect Antoni Gaudi’s work in light of other mathematical filters (i.e. mathematical inversion) and subsequently extracted large sets of data (2,000-3,000 images) to train a Generative Adversarial Network (GAN) for generating new visuals referencing the Sagrada Familia. This process capitalizes on one of several types of deep neural networks of the GAN family, first introduced by Ian Goodfellow in 2014, featuring competing networks for training generative models by unsupervised learning. (Goodfellow, et al., 2014) GANs, as described by Yann LeCunn, are “the most interesting idea in the last 10 years in Machine Learning”.

A typical GAN is defined by two independent deep neural networks that compete against each other: a discriminator network (D) and a generator network (G). The two networks compete against each other by engaging in an adversarial learning process of generating fake samples that are incrementally more realistic (generator) and a process of classifying fake samples from real samples (discriminator). While the generator network tries to predict features given a certain category, the discriminator network tries to predict a category given the feature of an instance of data.

The deep structure of such generator and discriminator networks allows more efficient learning process by adding more layers, where semantic learning is distributed. A deep learning network uses its hidden layer architecture, to learn incrementally categories that characterize low-level features like a mullion, then it learns gradually higher-level features like a frame and afterwards even higher features like a window. This is a major advantage of using deep learning networks, compared with other machine learning techniques that require domain expertise and intense feature extraction. Every node and neuron in a deep learning network describes one aspect of the image being learned and all nodes and neurons together provide the overall semantic representation of the image. The network’s nodes are adjusted using backpropagation, where after each forward-pass through the network, the backpropagation performs a backward-pass while adjusting the model’s weights. Backpropagation fine-tunes the weights of the network based on a loss function resulting from the previous training epoch. Lower loss function levels are ensured by properly tuning the weights of the network, which in turn improves model generalization.

In the case of GANs the optimization process is not aimed at finding a minimum (i.e. minimizing the loss function), but an equilibrium between the competing networks. The process of training two competing networks, (G) and (D) simultaneously, is inherently unstable, as the improvement of one network comes at the expense of the other network: “Training GANs consists in finding a Nash equilibrium to a two-player non-cooperative game. Each player wishes to minimize its own cost function […] unfortunately, finding Nash equilibria is a very difficult problem. Algorithms exist for specialized cases, but we are not aware of any that are feasible to apply to the GAN game, where the cost functions are non-convex, the parameters are continuous, and the parameter space is extremely high-dimensional” (Goodfellow, et al., 2016).

![Figure 1. Typical architecture of a GAN](image)

A statistical approach for architectural concept generation

The process tried to look into people’s own perception of what the building symbolizes in the collective unconscious or individually (a forest of columns or the “Gothic” style - in spite of Gaudi’s own regard of the building as Classical), and map these specific impressions on the built space visuals to generate versions of a mutated space based on these alternative realities. The Sagrada Familia was selected as case-study because the building demonstrates substantial spatial complexity, which manifests in a blurring of clear tectonic distinction of parts (Figure 2). For example, the typical architectonic paradigm of post and lintel, columns supporting beams, which in turn support slabs, gives...
way to branching columns which decrease in size, reaching out to support multiple combined vaults which compose the ceiling. As a result of the surface connection complexity, the semantic recognition of features by a neural network, is harder, especially in an unsupervised learning environment, where labeled data is not provided. Due to the complexity of the first domain we want to take advantage of robust process like machine learning which can examine thousands of samples. At the same time, the use of statistical method for visual results is no typical to designing, and makes outcome evaluation harder, as we discuss later.

Various interfaces were used for the respective stages of data preparation and training. Two original data-sets of Sagrada Familia interior perspectives were extracted and augmented from video files, to serve as primary data. These were trained separately with four data collections, each one combined with one of two Sagrada Familia training data-sets (SF1, SF2), to identify the response of the algorithm to image groups of different semantic expression (i.e. lines, specific geometric patterns, etc.), and therefore, their respective suitability for future training. Two data-sets contained about 3,000 unfiltered samples each from real life (Gothic cathedral images; Forest images), while the other two were generated using 3D modeling software.

1. Sagrada Familia image collection “SF1” (A) is trained with “Gothic cathedral” (B) images.
2. Sagrada Familia image collection “SF1” (A) is trained with “Forest” (B) images.
3. Sagrada Familia image collection “SF2” (A) is trained with mathematical Inversion Line (B) drawings.
4. Sagrada Familia image collection “SF2” (A) is trained with 3D surfaces (B) from Boolean subtraction.

Pytorch deep learning libraries were used within the Anaconda environment to augment the datasets and perform the training. For a better performing model, augmentation methods were applied by creating new, synthetic but reasonable examples from the initial input domains A and B. The size of image samples in the dataset was 512x512 pixels. We divided the dataset into 85% training and 15% testing sets for each A and B domains, using 2550 image samples for the training and 450 image samples for the testing.

Figure 2. Exterior-Interior views of the Sagrada Familia, showing the complexity of surface topology forming the window-ceiling connections (image credit: E. Vermisso).

Figure 3. Typical cycleGAN architecture showing the cycle consistency approach: the network has to prove it is not only capable of learning the domain translation from A to B but also from B to A; examples from four data-sets paired with respective examples from the Sagrada Familia data-sets.

Figure 4. Design workflow showing the software environments used to generate, process and feed data to the neural networks for training; two original Sagrada Familia data-sets were used for the separate training experiments (4 experiments total).
Results: Objective vs. Subjective Evaluation

What benchmark should we use for selection of “successful” results? Two criteria we can consider are how well the process of domain transfer is applied, and the visual quality of results relative to our specific problem search. The former is an objective factor while the latter is subjective. Can these factors be cumulatively considered? We should keep in mind that the best domain translation does not also guarantee the optimum solution for a design problem. Those outcomes which demonstrate a “good” workflow, like adequate transfer of desired features from domain A to B may not be the ones we favor in terms of other, more subjective requirements (i.e. aesthetics, novel configurations etc).

We isolated a few samples which exemplify the results’ diversity well (Fig.5). The “real A” and “fake A” samples are shown next to each other, for three domains: Gothic to SagradaFamilia, SagradaFamilia to Forest and Inversion to SagradaFamilia. Using metrics like “Structural Similarity Index” (SSIM) we can obtain some understanding of the results’ accuracy. However, these results seem to deviate substantially from human assessment: “While it is nearly effortless for humans to quickly assess the perceptual similarity between two images, the underlying processes are thought to be quite complex. Despite this, the most widely used perceptual metrics today, such as PSNR and SSIM, are simple, shallow functions, and fail to account for many nuances of human perception.” (Zhang, Isola, Efros, Shechtman, & Wang, 2018) Another metric used to evaluate the results uses VVG networks. We have annotated the “perceptual similarity” factors of our chosen samples, following those measured using SSIM (Fig.5). Can we reconcile objective and subjective means of assessment? At this point -at least for architecture- the use of automated evaluation should be complemented by human assessment: “One intuitive metric of performance can be obtained by having human annotators judge the visual quality of samples” (Goodfellow, et al., 2016). At the end of figure 5 we have manually annotated three samples we find interesting with respect to possible spatial transformations, indicating the directionality of the domain application based on the kind of features we feel were prioritized by the network.

The first example (epoch 15 of Gothic to SF) indicates some learning is already taking place and (G) is trying to apply semantic features from the real distribution of domain B (Gothic), like the perspectival effect of a Gothic church aisle to output a fake example of a more fluid space. The second example (epoch 33) uses a (real) input with similar perspectival point of view as the previous example, but the result is very different, indicating a bulging, or expansion of the vaulted ceiling, warping the columns around it. The reason may be the difference in the domain A (SF) example. Example 3 (epoch 16 of SF to Forest) is especially intriguing given the early training stage and the sample from the real “B” domain; it is only a patch of green, but the network managed to apply the intricate grass texture well on the vaulted ceiling to create a clear directionality of what looks like structural ribs. Overall, automated assessment clearly differs from human assessment.
While a network primarily looks at domain transfer without knowing the significance of specific semantic features, it may assign a low (good) value to a fake output if the transfer of a high-res texture or pattern is for example, successful, even if the image content seems it cannot stand up (may be rotated, or tilted). A designer would look at more features at the same time, like orientation or proportion to classify a result as successful. As a result, we feel that automated metrics at this point can be a complementary feedback to the more intuitive assessment necessary to filter results and revise the network architecture. Such metrics can include for example “Species Explorer” developed by Andy Lomas and Jon McCormack (2020).

Despite lacking a particular theory to provide a clearer trajectory of the experiment in advance, an intuitive result evaluation indicates that -given adequate training- the network can sample “Forest” examples and apply the kind of “branching” structure found in trees to the existing structure of the SF, rendering an even more organic architectural vocabulary. These imagined design alternatives can help architects expand their morphological/structural design repertoire by pushing the perception of what may be possible, while providing insights on the importance of specific architectonic elements based on the network’s reading.

**Improved results using high resolution data**

We should note that these experiments are an open-ended attempt to explore unsupervised learning as a heuristic workflow for a new conceptual understanding of already known semantic features of architectural space. The learned features which seem to be preferred and carried over by the neural network in the “hallucinated” output matrix may lead us to narrow training data-set towards “curating” specific results. After the workshop, the networks were further developed and trained to investigate outcomes of higher resolution (Fig.6) including larger datasets that better represented both domains (3000 samples of 1024x1024 each domain). The improved networks were trained for 240 epochs, using Adam optimizer with learning rate of 0.0002 for the first half of the training, and then reduced linearly to 0.0000 over the remaining iterations. We used a 512x512 input resolution, a batch size of 1 and a pool size of 50 to reduce model oscillation. (Shrivastava, et al., 2016) The model used a LSGAN mode and features a 70x70 PatchGAN discriminator architecture, a fully convolutional neural network looking at a patch of the input sample for which it outputs the probability of the sample being “real” or “fake”. It uses a resNet 9 blocks generator architecture; each layer is followed by an instance normalization and a ReLU layer. Resolutions of 256-512-740 were trained with batch sizes of 2-1-1 and pool size of 50-60-70. Figure 6 indicates improvement in the network’s ability to learn detailed feature representation from the input data and translate the semantic representation of a “Forest” domain to the “SF” domain. A remaining challenge, when working with GANs is the lack of objective metrics for evaluating network performance.

**Creative design potential in AI training**

It is interesting that the new samples created by the “generator” (G) network are labelled as “fake”; to us they are valuable, because they did not exist prior to the training procedure, and some of these manifest quite original, not typically considered configurations. Given the short time-frame of the initial experiment (2 days) we feel that an iterative process of training and revising the data-sets to achieve high quality training examples may increase both the fidelity of the results (resolution, appropriate semantic selection) as well as their potential for novelty, according to Boden’s criteria for assessing creative achievement: Novelty, Surprise, Value and Acceptability. (Boden 2004) Developing this work further, we can identify those examples that seem more promising with regard to a particular aspect of space, like the conception of a new structural system, or merely regarding morphological possibilities for surface manipulation. In this respect, these first outcomes, as well as their successors constitute a kind of creative outcome of the network. Still, without a particular goal in mind at the start of the process, it is hard to predict the outcomes, especially during later epochs, when results become more refined. Consequently, the value of the new images lies in their nature as outcomes of a “heuristic” process which has to be tested before assuming a meaning and enabling explanation. This type of “exploratory” process can potentially lead to innovation: “Exploration is the start of non-combinatorial creativity. Indeed, if the style of thought is an interesting one…then even just exploring it will lead to many novelties and may reasonably be regarded as ‘creative’”. (Boden, 2004) In the absence of a clear research question, a process of simulating “play” (open-ended investigation, Fig.7) can often engender high probability for creative output. Boden’s notion of P and H Creativity (psychological, historical) is not clearly applied anymore, as new
Figure 7. Heuristics usefulness for novelty, according to Boden

outcomes depend on an artificial neural network; while a designer may have encountered prior similar results, the network has not. With regards to any discussion on human-machine combined creativity, these relationships should probably be considered.

Within the Design domain, “...Newell and Simon (1972) describe a general theory of human problem solving”, where “...humans address problems by searching in problem spaces, where a problem space is defined by the goal and the domain knowledge, in the form of operators that enable the search...applied to design, the goals are part of the design requirements that express design variables and the ranges of values they can take. Some design theorists have characterized design creativity in terms of problem-space search: if the design variables and the ranges of values they can take remain fixed during design problem solving, the design is routine; if the design variables remain fixed but the ranges of values change, the design is innovative; and if the design variables and the range of values both change, the design is creative.” (Goel, 2019)

As previously mentioned, in a heuristic search for design solutions (like this) questions are somewhat open. By correlation to Newell and Simon, our search space is very large because the task is loosely defined (task = domain-transfer); we might say that the variables here are the semantic features which the network may recognize, and the values they assume are the weights of the particular layer in the network which range from -1 to 1. As we are not yet aware of the particular internal interactions of the network we cannot make an educated guess on the weight adjustment, however we know that the variables/features are not fixed, but depend on factors like the “quality” of the training examples (resolution, diversity, lack of bias). This structure of some degree of uncertainty, as defined in the Newell-Simon theory aligns with Boden’s claim of promise in heuristic processes, which we have here undertaken.

**Application Limitations: Extrapolation from Two to Three dimensions**

This experiment has assessed a domain translation process using GANs and identified relevant constraints. Training data is a critical factor which largely affects the network’s performance and the anticipated outcomes. The data used currently relies on recognition of 3-dimensional features from 2-dimensional images by an algorithm which does not understand it is looking at two-dimensional data. The network performance is also affected by how well the two domains are represented throughout the dataset. This can be observed in cases where a network learns a fair share of semantic characteristics and is able to perform successful domain translations from domain A to domain B but fails to perform translations from B to A. In this case, one of the domains is not represented well through the image samples, leading to unsuccessful translation in one direction. To ensure the trained model can generalize well, various augmentation methods were applied. (Hastie, Tibshirani, & Friedman, 2016) We observed that the network’s performance improves greatly if the dataset samples have some features consistent (i.e. background). Trained networks generalize well when the new input samples are reasonably different examples of the initial domains.

Can machines imagine space yet? The possibility for manipulating the third dimension to successfully generate appropriate training data remains a critical component in using AI for architecture. Deep Mind recently used a “GQN” algorithm (Generative Query Network, Eslami, et al., 2018), to solve the so called “Inverse Graphics Problem”. The network was fed a limited number of two-dimensional images of a three-dimensional space and was able to generate a new set of images of the space from a different viewpoint. Potentially, this can be used to generate 3D space from a handful of 2-dimensional inputs. While this seems promising for Architectural Design, the current output resolution is limited to simple spatial layouts (i.e. 3D shapes inside a room) because AI systems don’t yet fully understand the semantics of a scene (separation, background, foreground) to model complex configurations. In order to use this as a generative tool for “imagining” new architectural space (i.e. drawing one elevation of a building and asking the algorithm to generate another), we need to refine the network’s architecture.

**Conclusion**

Notwithstanding the evident practical applications of statistical learning in medicine, economics etc., consideration of corresponding implications for Design Thinking is critical. While machine learning re-emerged as a growing area of Artificial Intelligence during the last 15 years, it remains nascent within the architecture community. It is necessary to clarify the contribution of AI in the design disciplines, and assess the requirement for human input within collaborative human-AI workflows. The execution of this project in an architectural context poses some challenges in terms of technical skills; however, this kind of workshops can operate well, given more time. This can improve the investigation by providing the opportunity to discuss the particu-
lar design intent of the training, so the data is explicitly curated by the participants.

Furthermore, evaluation of AI-generated results is a challenging task and can happen in several ways; from a subjective perspective, any results of legible resolution which encompass an adequate amount of semantic recognition from the network can serve as guidelines to steer conceptual development of new surface morphologies or structural iterations based on the imagined alternatives of Gau- di’s work offered by the network. Technically, a challenge remains, of transferring diverse complex two-dimensional visuals into three-dimensional models. In certain cases, this may be hard; a secondary assessment is necessary to understand how to categorize and control learning of particular features, to output legible design possibilities.

From an automated process standpoint, adequate objective metrics also need to be incorporated, as current methods like SSIM and Perceptual Similarity factor are not very effective in sorting out what designers might consider “optimal” solutions. As we were interested in augmenting design ability at early conceptual stage, the objective metrics seem secondary -but not entirely unnecessary- at this point.

For future development, we can optimize both objective and subjective methods of evaluation. In relation to our own design intuition, we could ask a number of designers to select the “best” or most interesting results, cross-reference their preference and assess the diversity of the subjective evaluation. Then we can use additional tools for automating personal aesthetic judgment, like “Species Explorer”. This provides an interface to categorize computationally generated outcomes and identify evolutionary parameters which tend to give results similar to the designers’ programmed preference. (McCormack & Lomas, 2020)

From an architectural point of view, considering the current limitations of machine learning algorithms and the human bias embedded in their programming, it is fundamental to regard these outcomes as complementary, rather than antagonistic, to human thinking. By extension, we may regard the role of machine intelligence as positively disruptive, as early cyberneticists imagined it could be; Norbert Wiener’s chess-machine discussion mentions characteristically that “...it is unquestionably possible to construct a machine that will play chess in the sense of following the rules of the game, irrespective of the merit of the play. This is essentially no more difficult than the construction of a system of interlocking signals for a railway signal tower. The real problem is intermediate: to construct a machine which shall offer interesting opposition to a player at some one of the many levels at which human chess players find themselves.” (Wiener, 1965) Garry Kasparov identified the same limitation in early chess software, which demonstrated substantial fluctuation in skill, incorporating brilliant moves and serious errors in the same game. (Kasparov, 2017) Both Wiener and Kasparov express a humanistic perspective on human-machine interaction, identifying the assistive role of AI within the broader scope of human endeavor. People and Learning machines can reciprocally improve through interaction. Computa-

ACM Transactions on Interactive Intelligent Systems, Vol. 10, No. 4, Article 88. Publication date: October 2020.

Acknowledgements

The authors thank the “Gaudí’s Hallucinations” workshop participants (eCAADe/Sigradi 2019) for contributing to the data-set generation: Alexander Grasser, Begum Hamzaoglu, Daiki Kidô, Samim Mehdizadeh, Hongmei Wu.

References


Automatic Similarity Detection in LEGO Ducks

Mark Ferguson¹, Sebastian Deterding¹, Andreas Lieberoth², Marc Malmdorf Andersen², Sam Devlin³, Daniel Kudenko⁴ and James Alfred Walker¹

¹ University of York, UK. maf541@york.ac.uk, sebastian.deterding@york.ac.uk, james.walker@york.ac.uk
² Aarhus University, Denmark. andreas@edu.au.dk, mana@cas.au.dk
³ Microsoft Research Cambridge, UK. sam.devlin@microsoft.com
⁴ Leibniz University Hannover, Germany. kudenko@l3s.de

Abstract

The automated evaluation of creative products promises both good-and-scalable creativity assessments and new forms of visual analysis of whole corpora. Where creative works are not ‘born digital’, such automated evaluation requires fast and frugal ways of transforming them into data representations that can be meaningfully assessed with common creativity metrics like novelty. In this paper, we report the results of training a Spatiotemporal DeepInfomax Variational Autoencoder (STDIM-VAE) on a digital photo pool of 162 LEGO ducks to generate a phenotypical landscape of clusters of similar ducks and dissimilarity scores for individual ducks. Visual inspection suggests that our system produces plausible results from image pixels alone. We conclude that under certain conditions, STDIM-VAEs may provide fast and frugal ways of automatically assessing corpora of creative works.

Introduction

How to evaluate creativity is a major and ongoing concern in research, the creative industries, education, and many other areas. Researchers have developed numerous methods to assess both human and computational creativity across the “four P’s” – person, process, product, and press/environment (Kaufman, Plucker, and Baer 2008; Lamb, Brown, and Clarke 2018; Jordanous 2019; Plucker, Makel, and Qian 2019). In psychology, expert evaluations of creative products are often seen as the ‘gold standard’ (Plucker, Makel, and Qian 2019). In computational creativity evaluation, human judges are likewise frequently used (Lamb, Brown, and Clarke 2018). Expert evaluations feature comparatively high reliability, intersubjectivity, predictive value, and ecological and criterion validity: they are close to everyday practices around creative works like critiques, reviews, or prizes (Lamb, Brown, and Clarke 2018; Plucker, Makel, and Qian 2019). They also embody the sense-making within a social context that most contemporary definitions consider essential to creativity (Colton and Wiggins 2012; Plucker, Beghetto, and Dow 2004).

Automated Creativity Assessment

That said, expert evaluations are labour-intensive (Lamb, Brown, and Clarke 2018; Jordanous 2019); they don’t scale to the volumes of creative products one may find with cultural archives, generative systems, or large-scale testing. This has led researchers to trial computational formalisations of creativity such as novelty to automate creativity evaluation (Lamb, Brown, and Clarke 2018). One unique opportunity of automated evaluations is that they can provide rich statistical and visual analyses of a whole corpus (Grace et al. 2015; Elgammal and Saleh 2015), in addition to individual products. This makes them potentially akin to cultural analytics (Manovich 2016), the use of computational and visualisation techniques to analyse massive cultural data sets. A good example are expressive range analyses of procedural content generators (Summerville 2018), which visualise the frequency space of creative outputs of a system as a heat map.

While there have been some attempts to apply automated creativity evaluation to human works (Grace et al. 2015; Karampiperis, Koukourikos, and Koliopoulou 2014; Elgammal and Saleh 2015; Zhu, Xu, and Khot 2009), it remains chiefly confined to computational creativity (Lamb, Brown, and Clarke 2018). Human creativity assessment continues to use either poor-but-scalable self-reports and test batteries or good-but-expensive human expert evaluations (Kaufman, Plucker, and Baer 2008). We are missing robust, usable, validated computational tools for automatically evaluating human creative works. Such tools could not only provide cheaper, more reliable creativity measurement at scale: they would also allow us to analyse whole corpora of creative products in the style of cultural analytics.

The Context: The LEGO Duck Task

In response, we have been exploring the automated evaluation of a creativity exercise, the LEGO Duck Task (henceforth ‘Duck Task’). In this task, participants are instructed to make a duck from a standard set of six LEGO bricks. Task instructions can vary from e.g. making ‘the most creative duck’ to making as many different ducks as possible in a given time. The Duck Task has many attractive features for (automated) creativity assessment: It is easily understood across cultures. It is repeatable, unlike other task-based assessments where knowing the solution biases subsequent runs. Recombining the six bricks opens a vast phenotypic landscape of possible ducks and non-ducks. And yet the six bricks present a small set of simple, low-dimensional shapes that are relatively easy to formalise in terms of their (dis)similarity or other dimensions of interest.
One challenge we discovered early on is transforming physical LEGO ducks into computational representations. Standard methods of image recognition ran into interesting issues that are beyond the scope of this paper. We also considered but early on discarded the use of digital LEGO construction tools. Not only are physical LEGO bricks more accessible and familiar; research suggests that physical tools afford forms of embodied creative cognition that their digital remediations can constrain (Dove et al. 2017). We reasoned that this physical-to-data transformation poses a general challenge for in-the-wild automated evaluation of ‘born analog’ creative works. Hence, we began exploring potential ways of transforming large sets of physical works – LEGO ducks – into data representations that lend themselves to automated creativity assessment.

**Contribution & Structure of this Paper**

In this paper, we present one fast and frugal method for transforming a small corpus of physical creative works into a phenotypical landscape and individual novelty metrics using STDIM-VAEs, formalising novelty as corpus-relative dissimilarity. Our method generated plausible scores and clusters of human-meaningful similarity for our LEGO duck pool from raw pixels of simple mobile phone photos. This is particularly surprising, as existing methods for automatically assessing novelty rely on well-structured data sets in which human algorithm designers pre-specified likely meaningful dimensions (Grace et al. 2015; Pérez Y Pérez et al. 2011; Karampiperis, Koukourikos, and Koliopoulou 2014; Elgammal and Saleh 2015; Zhu, Xu, and Khot 2009; Correia et al. 2019).

We will first present the Duck Task and how we generated a diverse set of human-made LEGO ducks and photographed them. We will then present the computational architecture and methods we used to pre-process images, evaluate duck novelty, and generated a phenotypical landscape of the total corpus. Finally, we present our results and discuss their ramifications in light of the existing literature.

**Lego Duck Data Set**

A corpus of trial ducks was generated by passing members of the public during an open science event at Aarhus University. Sealed packs of the six bricks were piled on a tabletop under a large sign saying ‘Build a duck for science!’, displaying a yellow rubber duck but no finished LEGO ducks to avoid constraint by example. Participants who approached the booth were given a brief verbal introduction to the purpose and asked to build a duck in whatever way they saw fit, with various tweaks to the pattern throughout the day: sometimes encouraging builders to come up with something new, and many told to ‘just have fun’. Only after building a duck were participants invited to proceed to the back of the booth, where previous ducks were on display.

3D photography was conducted using an iPhone with the *Foldio 360 app* (orangemonkey.com/app) attached in place on a *Foldio 2* photography light tent, with a *Foldio 360* turntable for rotating ducks. Each duck photographed was stored as 36 individual jpg files, and as a 3D rotation video in .GIF and .mp4 formats. Later in the process, this was reduced to 24 pictures. 518 ducks were 3D photographed in the initial stages of the project, in groups of 169, 162 and 187 observations. The second group (162) was chosen for the study in this paper.

**Methodology**

**STDIM-VAE**

To assess the (dis)similarity of LEGO duck models, we compressed the high-dimensional image data into low-dimensional representations with the STDIM-VAE hybrid encoder (Ferguson et al. 2020). This combines features of a Spatiotemporal DeepInfomax (ST-DIM) (Anand et al. 2019) and a Variational Autoencoder (VAE) (Kingma and Welling 2013). The architecture for the hybrid encoder is shown in Figure 1.

![STDIM-VAE Network Architecture](image)

**Figure 1:** STDIM-VAE Network Architecture

As this encoder is designed for video data, the duck rotation videos were used, which stitch together the available images of each duck at different angles to generate a video with the camera ‘circling’ the duck. This is key since while the VAE section aims to purely encode visual information frame by frame, the ST-DIM section aims to learn a representation that maintains high mutual information between local features in the same spatial location in sequential frames, as well as between the global features and all the next sequential frame’s local features.

**Image Pre-processing**

Before feeding video frames into the network to train the encoder, we pre-processed them to generate a mask for each duck that would align images in each frame and reduce their size. Smaller images reduced the compute requirement, while alignment ensured dissimilarity is not reported due to shifts in the model placement. To generate masks, we first extracted edges using color-based edge detection (Chen and Chen 2010) and thresholding. We then applied Connected Component Labeling (CCL) (Wu, Otoo, and Shoshani 2005) to label each region. As the duck is fully contained within the image, any region that touches the side of the image can be discarded. A smoothing filter was then passed over the image along with a second round of thresholding. Next, we relabelled the resulting image. We then took the largest labelled region as the mask and: calculated its centre of mass,

---

1Data set available at https://osf.io/73kv2/.
recorded the distance from the centre to each mask edge, and determined a bounding box for the whole image set based on the maximum distance in each direction. This box was then used to crop each image around their centre point.

Evaluating (Dis)similarity

We combined three approaches to evaluate the (dis)similarity of LEGO duck models. Firstly, we used Uniform Manifold Approximation and Projection (UMAP) (McInnes et al. 2018) to project representations down into a 2D space. We then used hierarchical clustering with cosine distance over the concatenated representation of each frame. The extracted clusters were visually examined to access the similarity of ducks within the same cluster. Finally, each duck was ranked based on the average distance and cluster size (from hierarchical clustering). This allowed us to examine a small set of models that the system believes are either common or novel.

Results & Discussion

The projection generated by UMAP can be seen in Figure 2, where the point color represents the cluster label generated by hierarchical clustering.² Seven regions have been highlighted. While not strictly abiding by the labels from the hierarchical clustering, they visually appear to group. Visual inspection suggested that models within the same region indeed shared common traits to the human eye.

²Find an interactive version at https://osf.io/kyzum/.

Figure 2: UMAP of Lego duck representations

For ducks in region A (e.g. Duck 0,1), it appears that generally ducks have attempted to model the whole duck. Additionally, four parts of a duck’s anatomy are commonly created in the same manner. The head is commonly created by combining a 2 × 2 yellow brick on top of a 2 × 3 red plate. The tail is generally modeled by placing a 1 × 2 brick on the back of a 2 × 4 brick. The legs are commonly constructed by attaching a 2 × 2 brick under a 2 × 4 brick, although the position this attachment occurs changes within the region. Finally, a red 2 × 3 plate was often used to create the duck’s feet. In comparison, models within region C (e.g. Duck 97,109) appear to focus on modelling the ducks head, where it is common to use two 2 × 3 red plates for the duck’s bill. For region B, there appears to be no one common feature. Instead, the region seems to act as a transition between regions A and C. In region D (e.g. Duck 135,147), a duck’s wings are often represented by two 2 × 3 red plates on top of a 2 × 4 brick, although slight variations exist in how the duck’s head and tail are created. In the final two regions, F (e.g. Duck 34,57) and G (e.g. Duck 45,75), the duck’s head and feet are created in a similar manner to region A. However, models in regions F and G often differ based on the height of ducks. In the F region, the models normally have a maximum height of 3 bricks and 2 plates, whereas models in region G generally have a maximum height of 4 bricks and 1 or 2 plates.

When we visually examined the twenty clusters identified from hierarchical clustering, many of the clusters could indeed fit into one of seven regions previously discussed. Whenever a cluster did not clearly fit into one region, the cluster was normally small and contained models that visually looked novel.

Finally, the ducks were ranked based on average distance to all other ducks and the size of the cluster. Common ducks should have a low average distance and large cluster size, whereas novel ducks should have a high average distance and small cluster size. The top two models for each of these are shown in Figure 3.

Figure 3: Common ducks identified with (a) small average distances and (b) large cluster sizes. Novel ducks identified with (c) large average distances and (d) small cluster sizes

We assume our method produced plausible results at least partially because the possibility space of creative works (LEGO ducks) and resulting pixel distributions is well-constrained and features inherent segmentation in the shape of LEGO blocks. That is, the informational properties of our raw data offloaded some of the ‘heavy lifting’ of recognising meaningful (dis)similarities, which human algorithm designers may need to do with more diverse corpora by pre-specifying semantically rich features or dimensions. This was further aided by our image pre-processing, cropping and aligning all image data. Any remaining dissimilarities were likely to be ‘inherently’ meaningful and structured. Hence, we would not expect our method to easily generalise to more inherently diverse corpora like e.g. ‘construct any entity from any kind and number of LEGO bricks’. However, for our purposes of developing an automated creativity evaluator of a scalable human creativity assessment – the Duck Task – the initial results are encouraging.
Conclusions & Future Work

The main aim of this work was to test the viability of a STDIM-VAE on photo imagery to create a representation that allows the easy creativity evaluation of products of the Duck Task human creativity assessment. The UMAP projection generated from our data shows that the representation indeed encapsulates human-legible feature differences in duck models, such as using particular bricks/plates to create particular parts of duck anatomy, or modelling the whole duck vs. just the head. Ranking ducks on two dissimilarity metrics generated a ranking topped by ducks that appeared on first inspection to be novel.

To validate our findings, future work will compare these to rankings by human expert evaluators. Further experiments on unused datasets would allow to test the replicability of our method, including evaluating the trained encoder on duck models unseen during training. Additionally, by sweeping all starting points to find the best match, the assumption of temporal alignment can be removed. Finally, exploring the generalisability of our approach to more diverse corpora is an interesting area for future work.

Acknowledgments

This work was supported by Microsoft Research PhD Scholarship Programme, the Digital Creativity Labs funded by EPSRC/AHRC/Innovate UK (EP/M023265/1), and the PLAYTrack project, supported by a research grant from the LEGO Foundation.

References


Evolutionary Experiments in Typesetting of Letterpress-Inspired Posters

Sérgio Rebelo¹ and João Bicker¹ and Penousal Machado¹
¹University of Coimbra, CISUC, DEI
{srebelo, bicker, machado}@dei.uc.pt

Abstract

In this paper, we present a system that generates letterpress-inspired typographic posters from a user-given text. This system employs lexicon-based approaches not only to recognise emotions and sentiments but also the colours related to the content. Moreover, the system generates and evolves outputs employing an Evolutionary Computation approach. The generated outputs are evaluated according to their legibility, aesthetics and semantics, throughout a multi-criteria hardwired fitness function scheme. Also, this system enables its users to guide the generation process through the dynamic definition of several setting parameters.

Introduction

Visual artefacts for public proclamation, i.e. posters, were already present in ancient societies and, over the centuries, they adapted themselves to the new social and technological contexts. Nowadays, they still maintain a palpable presence in our cities’ landscapes (Guffey, 2014). The recent advances in computational tools, especially on Artificial Intelligence (AI), are promoting deep social changes and, consequently, graphic designers began to design posters exploring these innovative technologies in order to create more interactive and immersive artefacts.

In this paper, we present a work in progress system that generates letterpress-inspired typographic posters from a text inputted by the user. Letterpress is a printing technique that became popular on the follow-up of the industrial revolution because it allowed a cheaper, easier and faster printing of commercial posters for mass communication (Guffey, 2014; Meggs and Purvis, 2016). Nevertheless, the design of these posters was slightly different from the typical contemporary design process. At the time, letterpress designers created posters while trying to fill a matrix, often in collaboration with the client. The design decisions were very pragmatic: extensive sentences were composed in condensed typefaces, and short sentences were composed in extended typefaces. Also, the content’s most important parts were emphasised through the use of bigger typefaces (Meggs and Purvis, 2016).

The present system generates outputs using a workflow, similar to the letterpress design process, based on our previous work (Rebelo et al., 2018). Briefly, the system (i.e. the designer) composes the content, inputted by the user (i.e. the client), dividing it into text boxes in order to fulfil, as much as possible, the posters’ canvas (i.e. the matrix). In this process, lexicon-based approaches are employed on the content to recognise sentiments, emotions, and colours related to the text. Also, an Evolutionary Computation (EC) approach is employed to create and evolve a population of poster designs. The generated posters are evaluated according to its (I) legibility, (II) aesthetics, and (III) semantics. The fitness of each output is assigned by a multi-criteria hardwired fitness function scheme. Also, like in the traditional letterpress process, the users may guide the generative process. Thus, the users may communicate their preferences to the system by defining, and iteratively updating, several setting parameters in a dedicated interface. This system is available online at https://pf.dei.uc.pt/eetlp.

System Overview

The system is a web application that generates posters through the employment of 3 main modules: (I) Input Processing; (II) Evolution; and (III) Evaluation. The Input Processing module employs lexicon-based approaches to recognise sentiments, emotions and colours related to the text. The Evolution module randomly initialises a new population of candidate solutions (i.e. poster designs) and uses a Genetic Algorithm (GA) to evolve this population. The Evaluation module assigns the fitness of each poster through a multi-criteria hardwired fitness assignment scheme. The fitness is calculated based on 3 criteria: (I) legibility; (II) aesthetics; and (III) semantics. A schematic of the system is overviewed in Figure 1.

Through a dedicated interface, the users can guide the system’s generative process by defining several settings parameters. The parameters are (I) the weights of each criterion in the fitness assignment scheme, (II) the GA’s setup parameters (i.e. the number of generations, the population and elite sizes, and the mutation probability), (III) the visual proprieties of the outputs (i.e. the posters dimensions, the number of rows in the grid, the sizes of the margins, the visual centre offset, and the optimal percentage of white space), (IV) the available colours, and (V) the typeface and its available weights. These parameters may be modified at any time during the generation. This way, during the generation, users can guide the system, adjusting these parameters.
Also, users may export the outputs at any time during the evolution.

Figure 1: Schematic of the system’s architecture.

**Input Processing Module**

The Input Processing module analyses the text, inputted by the user, to recognise the sentiments, emotions and colours related to it. This module is implemented by using the natural language facility library *Natural.js* (Umbel, Ellis, and Mull, 2020) and the lexical database *WordNet* (Fellbaum, 1998).

First, this module subdivides the text into lines, using a sentence tokenizer. After, it performs a lexicon-based analysis on the words of the text. Thus, after tokenizing and lemmatising the input text, each word is searched in a word-emotion association lexicon developed by Mohammad and Turney (2012). This lexicon enables the recognition of 8 basic and prototypical emotions (i.e. anger, anticipation, disgust, fear, joy, sadness and surprise) and 2 sentiments (i.e. positive and negative) in about 15000 English words. Thus, it perceives what are the parts of the text with more emotional and sentimental charge.

The present module also performs an analysis to recognise the intensity of the relation between the colours and the text. This analysis is performed using a word-colour association lexicon developed by Mohammad (2011). This lexicon scores the intensity of the relation of 14000 English words with 11 colours: black; blue; brown; green; grey; orange; purple; pink; red; white; and yellow. In the end, the module creates a list that describes the intensity of the relation of each one of the 11 colours with the text. This list is sorted by intensity. The intensity of a colour is the sum of the scores obtained whenever this colour is associated with a word in the text.

**Evolution Module**

The Evolution module implements a GA to create a population of poster designs at random and, subsequently, evolve them employing variation operators, i.e. crossover and mutation, on the most promising outputs. The posters are selected by tournament based on their fitness. This method practices elitism, persevering the best individual of each generation to the next generation.

Each poster is a set of arranged text boxes. The text boxes are encoded as a sequence of arrays of numbers (i.e. the genotype). The first array in the sequence is a one-position array that encodes the poster’s typography colour (i.e. the colour configuration gene). The following arrays encode the text boxes (i.e. the text boxes genes). Each text box gene is composed of 3 numbers encoding the font’s weight, the text box height, and the font size in percentages of the height, respectively. Since the content of the posters may have different lengths, the number of text boxes and, so, the size of genotype may vary. The posters’ canvas is subdivided in a one-column grid with multiple rows that constraint the text boxes position and sizes. Perceptible poster designs (i.e. phenotypes) are generated through the rendering of the text boxes, according to the settings encoded in genotype.

**Initialisation**

The initialisation method generates a population of poster designs at random. Thus, for each individual in the population, it defines the colour configuration gene by randomly assigning one colour from the range of options available. The number of text boxes is defined by the number of lines of the text (i.e. each text box contains a line). The proprieties of each box are defined as follows. The font’s weight is randomly selected according to the range of options available for the selected typeface. The height of the text boxes, although selected at random, is defined by ensuring that the text boxes fill all the available space on the canvas. We ensured this by randomly generating a sequence of numbers with the same length of the number of text lines and the sum equal to the number of rows of the grid. Next, this sequence is shuffled and each number is assigned to a text box. The font size is always defined at 100% of the height.

**Variation Operators**

Poster designs are evolved iteratively, through the employment of crossover and mutation operators. Both operators are designed to preserve the validity of the generated individuals.

The crossover operator generates new poster designs through the exchange of genes between two parents. This way, it randomly selects two parents regarding their fitness and, thereafter, employs a uniform crossover method, which randomly selects which of the parent will give the gene to the children. This operator does not crossover the genes related to the height of the text boxes, ensuring that the generated children fulfill all the available space on the canvas.

The mutator operators perform random modifications in some parts of the individuals’ genotype. We designed these operators by ensuring that they covered all the search space. This resulted in 2 operators: *Independent*; and *Swap*. The mutations are performed based on a certain probability. Thus, the system for each candidate solution in the new offspring randomly defines if it will be mutated and, next, selects the mutation operator. Each operator has the same probability of being selected. The *Independent mutation* operator randomly selects a gene and, subsequently, randomly selects a parameter in the gene for the mutation. Each type of parameter has its own mutation method. If the colour configuration gene or the font’s weight parameter of a text box are selected, it randomly assigns a value to it according to the options available (i.e. number of available colours or weights). If the text box’s height parameter is selected, two genes are randomly selected, having one, at least, the height value bigger than one. After, it decides what will be the gene
that will decrease the height and the one that will increase. This selection is performed randomly unless one of the selected genes have the value 1. In this case, the gene with the value 1 will increase its height and the other will decrease. Finally, when the font size parameter is selected, it decreases or increases this value in 1%. The direction of this mutation is randomly calculated unless the value is already in its maximum threshold, \textit{i.e.} 100\% (the value will only be decreased) or in its minimum threshold, \textit{i.e.} 30\% (the value will only be increased). On the other hand, the \textit{Swap mutation} operator, as the name indicates, randomly selects two text boxes, in the same individual, and swaps the value of their genes.

\textbf{Evaluation Module}

The Evaluation module implements a multi-criteria hard-wired fitness assignment scheme to evaluate the outputs. This way, the outputs are assessed according to 3 criteria: (I) legibility, \textit{i.e.} how much content it is possible to read on the poster; (II) aesthetics, \textit{i.e.} how much the design of the poster satisfies a set of aesthetics measures; and (III) semantics, \textit{i.e.} how much the poster visual characteristics convey the semantic meaning of its content. Each criterion has its evaluation method. The fitness of each poster is calculated by the weighted arithmetic mean of the legibility, aesthetics and semantics. The weight of each criterion is defined by the user.

\textbf{Legibility} The legibility objective measures how much of the content is legible on the poster. The legibility of each text box is the difference between its target width (\textit{i.e.} the posters available width) and the width of its content when rendered. A text box’s content should always be rendered inside of the text box (\textit{i.e.} target width) and the white space, inside of the text box space, should be minimised as much as possible. This way, the legibility of a text box is the difference between the target width and the width of content when rendered. This difference is after being mapped to assign a poor assessment when the rendered text exceeds the size of the poster and to prejudice the text boxes that surpass a certain amount of white space. The overall legibility value of a poster is the weighted arithmetic mean of the value of text boxes. The weight of each text box in the mean is defined based on its height.

\textbf{Aesthetics} The aesthetics objective measures how much the design of the poster satisfies a set of aesthetic measures for the design of typographic posters. These measures are based on the work of Harrington et al. (2004). Nevertheless, they were adapted to the context of this work. This way, the aesthetics of a poster is evaluated according to (I) the alignment, (II) the regularity, (III) the balance, (IV) the white-space fraction, and (V) the composition security. The overall aesthetic measure is the arithmetic mean of these attributes.

The alignment attribute measures how regular is the horizontal placement of the text boxes on a poster. Thus, the module compares the distance between the values of the vertical positions of the left edges of the neighbouring text boxes. The closer the vertical distance between text boxes, the higher is the alignment score. The overall alignment measure is the arithmetic mean of all distances.

The regularity attribute measures how regular is the vertical placement of the text boxes on a poster. The calculation of the regularity is similar to the calculation of the alignment. However, it compares the positions of the top edges instead of the left edges.

The balance attribute measures how much the poster is centrally balanced. The centre balance of a poster is the difference between the centre of its visual weight and its visual centre. The centre of the visual weight of a poster is calculated based on the visual weight of its text boxes. The visual weight of a text box is defined by its area times its optical density. After calculating the visual weight of all text boxes, the overall measure of balance is calculated according to the method purposed by Harrington et al. (2004).

The white space fraction attribute measures if the percentage of white space in the poster is according to a certain optimal percentage threshold. This way, the overall measure is the absolute value of the difference between the current percentage of white space and the optimal percentage.

The composition security attribute measures if the text boxes positioned near the edges of the poster are secure and do not appear to fall off. The security of each text box is the minimum between the top and bottom edges. The overall value is the minimum between the values of all the text boxes.

\textbf{Semantics} The aesthetics objective measures how much of the posters’ visual characteristics convey the semantic meaning of its content. This way, the most important parts of the content should be emphasised in the layout, over the less important ones, and the typography colour should be related to the content. The semantics of a poster is, then, evaluated according to (I) the typography colour, and (II) the layout of the text boxes. The overall semantic measure is the arithmetic mean of these attributes.

The colour employed on the typography conveys the semantic meaning of the poster’s content when it is the colour most related to the content, according to the results of the analysis performed by the Input Processing module. This way, the overall value of the appropriateness of a typographic colour is the distance between the current used typographic colour and the most related colour to the content. This distance is calculated based on the content-colour intensity list defined before. When the most related colour is not available, the module considers that the most related colour is the available colour following on this sorted list.

The layout of the text boxes should emphasise the most important parts of the content by assigning them higher text boxes. We consider that the most important text boxes are those with a higher amount of emotion and sentiments recognised in the analysis performed by the Input Processing module. This way, the module defines the weight of one emotion on the poster by dividing the total number of emotions and sentiment recognised by the number of rows of the grid. After, it defines the optimal height for each one of the text boxes on the poster by multiplying the number of recognised emotions, in each text box, by the emotional weight calculated before. The appropriateness of a text box’s layout is, then, the distance between its current height and its...
optimal height. The overall value of the appropriateness of a layout is the arithmetic mean of all distances.

**Discussion and Conclusions**

We conducted a set of preliminary experiments to study and analyse the possibilities of the system evolving posters for several contents with several lengths and textual purposes. The experimental parameters used in these experiments were defined by empirical exploration and summarised in table 1. The weights of the criteria in the fitness assignment scheme were 90% for legibility, 5% for aesthetics and 5% for semantics. These weights were also defined by empirical exploration. The experiments were conducted using 3 typographic families published by Font Bureau and available at Adobe Typekit service: (I) Bureau Grot; (II) Titling Gothic FB; and (III) Benton Modern Display. In these experiments, the system used all the available colours and typefaces weights.

**Table 1: Experimental parameters.**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generations</td>
<td>300</td>
</tr>
<tr>
<td>Population size</td>
<td>20</td>
</tr>
<tr>
<td>Elite size</td>
<td>1</td>
</tr>
<tr>
<td>Mutation probability</td>
<td>0.7</td>
</tr>
<tr>
<td>Phenotype size</td>
<td>$298 \times 420$</td>
</tr>
<tr>
<td>Margin size</td>
<td>15 px</td>
</tr>
<tr>
<td>Grid</td>
<td>$26 \times 1$</td>
</tr>
<tr>
<td>Optimal percent of white space</td>
<td>50 %</td>
</tr>
</tbody>
</table>

Figure 2 display some results. More results are available at https://cdv.dei.uc.pt/evoposter/. Visually observing the results, one can conclude that the system can generate posters that achieve a high level of diversity and variation in terms of layout and colours. We also observed that the system, for the same text and under the same settings, generates results that although share similar visual characteristics are not identical. However, the diversity of the results is directly related to the distribution of emotions on the text and/or the strength of the relation of some words with colour. Texts with a uniform or weak emotional and sentimental charge and/or weak relation with colours tend to generate more diverse outputs.

Besides its capability to generate posters from scratch, we also observed that the system is a functional co-creativity tool. We believe that this system is a useful tool for enhancing users creativity (mostly graphic designers) when they design posters, especially in the earlier and most exploratory stages of their design processes. Also, it reveals the potential that AI techniques may have in the future practice of GD, mainly EC. Future work on this system will focus on (I) exploring different fitness assignment schemes, (II) using Natural Language Understanding methods to create more reliable textual analyses, and (III) including images and illustration on outputs.

**Acknowledgments**

This work is partially supported by national funds through the Foundation for Science and Technology (FCT), Portugal, within the scope of the project UID/CEC/00326/2019 and under the grant SFRH/BD/132728/2017.

**References**


3. Language and Narrative
Rosetta Code: Improv in Any Language

Piotr Mirowski¹, Kory Mathewson¹, Boyd Branch¹,²
Thomas Winters¹,³, Ben Verhoeven¹,⁴, Jenny Elfving¹

¹Improbotics (https://improbotics.org)
²University of Kent, United Kingdom
³KU Leuven, Dept. of Computer Science; Leuven.AI, Belgium
⁴ERLNMYR, Belgium

Abstract

Rosetta Code provides improv theatre performers with artificial intelligence (AI)-based technology to perform shows understandable across many different languages. We combine speech recognition, improv chatbots and language translation tools to enable improvisers to communicate with each other while being understood—or comically misunderstood—by multilingual audiences. We describe the technology underlying Rosetta Code, detailing the speech recognition, machine translation, text generation and text-to-speech subsystems. We then describe scene structures that feature the system in performances in multilingual shows (9 languages). We provide evaluative feedback from performers, audiences, and critics. From this feedback, we draw analogies between surrealism, absurdism, and multilingual AI improv. Rosetta Code creates a new form of language-based absurdist improv. The performance remains ephemeral and performers of different languages can express themselves and their culture while accommodating the linguistic diversity of audiences.

Introduction

Theatre is one of the most important tools we have for sharing experiences and building cross-cultural understanding. Moreover, theatre performers and audiences who speak different languages are more connected than ever, thanks to increasing ease of communication, dissemination of culture, translation, travel, and improvements in remote performance capabilities. In particular, improvised theatre (improv) is well positioned to connect culture given its universality, accessibility, and low barriers to entry: improvisation techniques can be readily understood and internalized, and in a short manner of time, individuals from diverse cultures empathize with each other while performing scenes together, with deep characters, relationships, settings, motivations, and subtext. Improv serves as a microcosm of cultural communication; it is “the theatre of the people” in moment (Boal 2006). Improv is therefore an ideal test-bed to explore broad cultural and communication questions (Mathewson 2019).

Improv is also a paradoxical cultural artifact. On one hand, improv is ubiquitous and conveys universal messages about the human condition and the vagaries of life. On the other hand, as a highly linguistic art form, improv is nearly impossible to understand if you do not know the language in which it is performed. Given that improv is based on the connection between the audience and the performers, watching improv in a foreign language severely limits this link. This contrasts with scripted theatre, which has been salvaged from monolingual oblivion: Sophocles, Shakespeare, and Sartre continue to be translated into many different languages, reinterpreted, and enjoyed by audiences around the world. Improv has not had such an opportunity, and performance groups are bound to remain local or switch to English as a lingua franca when performing internationally.

The art of improvisation is derived from the connections between performative layers, both between the performers, and between the performers and the audience. Improv embraces the audience to create collaboratively together. In this way improvisation is a democratic narrative, and the potential impacts of improvised theatre between performers and audiences of different cultures and languages are significant. Most international improvisational collaboration is English based, but many regional festivals take place in the languages of the host region. These performances exclude audiences without knowledge of the performance language, and limits the contributions of improvisors who do not speak the language. Without translation, improvisation misses out on important voices due to language limitations.
How can we create conditions so that improvisors from different cultures can improvise together in their own language? How can audiences understand performers using different languages? How might we grow our cultural communication and empathy while only being able to speak one language? Rosetta Code answers these questions, and gives theatrical improv a suite of software, scenes, and show structures from which to advance and expand.

The Methods section describes the technical details of the system, the challenges associated with improv in any language, and how we used our system in the context of theatrical improvisation. In section Rosetta Code on Stage we provide results of using the system in three shows using nine languages. We also present evaluative feedback from performers, audiences, and critics. In section Related Work, Historical Context and Discussion we situate Rosetta Code at the intersection of improvisational theatre and language, present an exploration of the cultural importance of multi-lingual artistic performance, and provide several directions for future work.

Methods

Artificial intelligence-based improvisation is an art form, where a robot and/or AI is used on stage as an improv stage partner (Bruce et al. 2000; Mathewson and Mirowski 2017a; 2017b; Mirowski and Mathewson 2019; Jacob et al. 2019; Winters and Mathewson 2019; Liu et al. 2019; Mathewson 2019). That robot relies on a generative language model to produce lines or actions in response to context, and can in itself be seen as a computationally creative system. A variation of that format, Improbotics, designed in 2016, consists in letting human actors enunciate the lines: the chatbot effectively whispers lines into the ears of human improvisers, who are only allowed to repeat exactly those lines, but are otherwise free to express themselves with a full vocabulary of physicality and emotions (Mathewson and Mirowski 2018). We have adopted this configuration for the Rosetta Code show.

The core idea of Rosetta Code is to build on existing, state-of-the-art language technology (Section: Technology Overview) to enable a palette of improvisational games (Section: Improv Games). Rosetta Code thereby allows improvisors speaking different languages to perform multilingual improv theatre together on stage.

Technology Overview

The technical setup used in this project consists of several elements that can be seen as independent building blocks, each corresponding to a piece of equipment or to an Application Programming Interface (API).

• Speech recognition (e.g. Google Speech-to-text API1 or Web Speech API2), which works in multiple languages3, running in a browser application. In order to successfully capture the improviser’s voice while occluding ambient noise and other performers’ voices, we rely on handheld dynamic vocal microphones (with an on-off button that can be triggered by the user), connected to the computer via an analog-to-digital audio interface.

• An instantaneous translation system, e.g. Google Translate API4, is used as communication channel to convert recognised speech from one language into another.

• A surtitle visualization interface (Figure 2) that enables the audience to follow the conversation using instantaneous translation, and allows improvisors to modify translation language settings.

• Text-to-speech synthesis API to automatically voice translations.

• In-ear headphone interfaces (or earpieces) enable individual performers to listen to audio translation while still being able to follow other conversations. Our setup to transmit sound from the computer to the improviser relies on FM radio transmitters that can multicast information to multiple FM radio receivers worn by several improvisers.

Improvisational Chatbot System

To respond meaningfully to human improvisor input utterances, the AI improv system works by using a statistical language model to generate sentences in continuation of some context presented as text. Previous versions of AI improvisation were built upon the neural network sequence-to-sequence architecture (Sutskever, Vinyals, and Le 2014) trained on a pseudo-translation task from the context into the generated output (Vinyals and Le 2015). For Rosetta Code, we rely on the GPT-2 neural network transformer architecture (Radford et al. 2019), trained on a large corpus of web pages, which we fine-tuned on the OpenSubtitles corpus5 of film subtitles (Tiedemann 2009).

1 https://cloud.google.com/speech-to-text
3 https://cloud.google.com/speech-to-text/docs/languages
4 https://cloud.google.com/translate/docs
5 https://opensubtitles.org/
Figure 3: Example of performed translation scene. The performer downstage speaks into the microphone in Dutch. Translation into English appears on screen and is fed via ear-piece into the ear of the performer upstage.

It is straightforward to integrate any existing chatbot into the Rosetta Code system, by replacing the machine translation component by that chatbot component. The chatbot acts like a sort of translation from one language into that same language, one sentence later. The virtual AI improv chatbot controlling a human performer can be seen as a difficult stage partner whose language often veers on the absurd and forces improvisers to resort to nonverbal communication (Mathewson and Mirowski 2018; Mathewson 2019).

Improv Games

There are several improv games that can be played by multilingual improvisers using the aforementioned technology. These games can be subdivided into translation-free games (that do not require the translation service), translation-based games, and vocalisation-based games.

In the following descriptions, we refer to the primary language spoken by the audience as the majority language and to the other, “foreign”, languages spoken by the improvisers as minority languages. We also make several assumptions about the improvisers, the audience, and the languages they speak. First, we assume that some improvisers speak only one (majority) language, while others master or have working knowledge of multiple (majority and minority) languages. Second, we also account for some members of the audience being fluent in multiple spoken languages. These configurations enable different combinations of information asymmetry during the show.

Translation-free Games There are several different improv games we propose for playing improv with improvisers speaking different languages. For this, we build upon gibberish improv games, in which one or more of the performers speak in a non-existing language (Johnstone 1979). However, in our games, each performer is allowed to express themselves by formulating real language. This setup is not unlike the vision of absurdist playwrights such as Samuel Beckett or Eugene Ionesco. They see language as being purely aesthetic (in our case, each utterance is fully formulated) and devoid of semantic significance (in our case, most performers and audience cannot understand the minority languages): “the Theatre of the Absurd shows the world as an incomprehensible place” (Esslin 1960).

Specifically, we propose to adapt existing improv exercises to the following games:

- **Stranger in a Strange Land**: a stranger does not understand the language of the others, and the others do not understand the language of the stranger either. This game can be played when several minority language performers (e.g., Chinese speakers in an European country) improvise with a majority language performer and for a majority language audience (e.g., English), thereby reversing the usual majority-minority status relationships faced by minority language speakers.

- **Languages of Love**: two performers are on a blind date or in a long-term relationship, but clearly speak a different language. This is a setup that invites the performers to seek equal status.

- **Tower of Babel**: Every performer speaks a different, unique, minority language, and does not understand the language of the others.

These games were designed to investigate how meaning and understanding can emerge without words, and force the performers to explore alternative means of communication, via body language, signalling, and bold assumptions. This relies on the assumption that they share some cultural and social references. These games can be adapted to reinstate partial information flow and understanding from some performers, by making one of the performers speak in a majority language. For instance, a variation of Languages of Love or Tower of Babel pairs majority language performers with minority language performers, and all performers assume they fully understand every one else, creating opportunity for comedic confusion if this is not the case.

Translation-based Games We propose the following translation-based games that we have devised, and which rely on speech recognition, live translation, and a combination of earpieces for performers and surtitles for the audience.

- **Lost in Translation**: this game is the translation-based equivalent of Stranger in a Strange Land: a minority language improviser speaks in their own language, with live translation. While the audience can read the surtitles, the remaining majority language improvisers cannot. The status and comedy of this game stem from allowing minority language performers to be understood by the audience while majority language performers struggle to make educated guesses about the meaning of the scene (Fig. 3).

- **Foreign Film**: every performer speaks the same minority language, and the majority language audience sees the subtitled translation of the scene, thus being able to connect with the story told by minority language players.

- **Babelfish**: combining ideas from both above games, we allow one or several minority language improvisers to
speak in that language while live translation is simultane-
ously shown to the audience (via surtitles) and sent to ma-
jority language improvisers (via earpieces). Everyone can
understand everyone else - albeit with time delays and er-
rors due to speech recognition errors combined with ma-
chine translation errors.

Vocalisation-based Translation Games Finally, and
rather than focusing on multilingual understanding, we go
back to linguistic experiments – dear to the Surrealists –
done purely on the sound of words, and we adapted two ex-
tisting translation-based games inspired by Raymond Que-
neau’s “Poor lay Zanglay” – a seemingly gibberish text
that makes sense in French when read aloud by an English
speaker, taken from *Exercices de Style* (Queneau 1947).

• **Telephone Game:** in this simplest of translation-based
games, performer A whispers a phrase in their minority
language (that only A can speak) to performer B on their
right. Performer B then tries to repeat, as well as possible,
what they heard to C, who whispers it in turn to D, and so
on. At the end of the game, both A and the last performer
speak the phrase out loud to the audience and into a trans-
slation system, and the original utterance is compared with
its repeated distortion.

• **Diplomat:** in this variation of Telephone Game, a major-
ity language performer needs to deliver a full speech in
a minority language that they do not know. They receive
that speech via an earpiece and what they say is translated
into majority language via live translation.

While such a list is far from being exhaustive, we believe
it exhibits a wide variety in the amount of information that
can be transmitted between the performers and from per-
formers to the audience. The varying ratios of information
asymmetry and misunderstanding thus create multiple op-
portunities for comedy.

Structure of a Multilingual Improv Performance

Using the technology and interface described above, along
with the various games we have devised, we came up
with the following script for a language technology-enabled
scene-based improvised comedy show:

1. Part 1: Human Miscommunication:
   (a) **Telephone Game,**
   (b) **Languages of Love,**
   (c) **Tower of Babel,** followed by a replay of that same scene
      in the majority language, and
   (d) **Diplomat** or a scene with two improvisers trying to
      perform in a relatively well-known minority language
      (e.g., French in the UK).

2. Part 2: Machine Translation:
   (a) **Lost in Translation,**
   (b) **Foreign Film,** and
   (c) several language combinations in **Babelfish.**

3. Revelation: the AI behind machine translation takes over.


As the last two items indicate, we have created a narrative
arc in the structure of the show: namely, we add a revela-
tion where the AI tools used for speech recognition, ma-
chine translation and text-to-speech end up getting a life of
their own and taking over the show.

This revelation serves two purposes. The first one is to re-
mind the audience that today’s multilingual communication
is enabled by machines, specifically by pattern recognition-
based AI algorithms, that present their own limitations and
sources of errors: the machine take-over in the show serves
as metaphor for the rise of imperfect AI in mediating human
communication, and the increased risk for miscommunica-
tion and misunderstanding due to algorithmic errors. The
second one is, as we describe in the following section, a re-
turn to the absurdist roots of our language-based games; by
delegating some of the language generation in improvised
scenes to a machine, we force the performers to find and
create meaning outside of the realms of verbal communica-
tion.

Experiments with Automatic Translation

In parallel with the development of Rosetta Code, we piloted
basic interaction-based experiments with automatic transla-
tion, the most significant being the *automatic translation to-
ward meaninglessness (ATTM).* The ATTM system allows
for automatic homophonic translation, also known as allo-
graphic translation or transphonation. The origins of this
linguistic genre can be traced back to at least 1450, with the
English-Latin transphonation “Mare eate ootys”, now more
colloquially known as “Mairzy Doats” (Kington-Oliphart
1886; Opie 1952). ATTM is a digital interface between a hu-
man and a web-enabled computer serving a webpage. The
automatic process works as follows:

1. ATTM takes as input any single line of text from a hu-
man interacting with it. For instance, the human might
say: “the sun sets behind the mountain, as the snow re-
vents.” ATTM’s speech recognition system would at-
tempt to convert the captured audio inputs to text. The
recognition is not perfect and the minor errors are where
the beauty of ATTM stem from.

2. ATTM then uses text-to-speech to synthesize the text to
audio which is played over the computer’s speakers.

3. While ATTM is synthesizing the new line, it is simulta-
neously listening with the microphone to the synthesized
sound. That is, it is listening to itself speak and attempting
to recognize its own words.

4. ATTM loops forever.

ATTM can speak to itself in this endless, yet continuously
degrading, loop. There is a minor modification which in-
roduces the difficulties of multiple language understanding.
Rather than setting the system to recognize the spoken text
as an English sentence, it instead recognizes the spoken au-
dio as if it were a French speaker saying a French sentence.
Obviously, the English sentence doesn’t sound like a French
sentence, but the system does its best to recognize the words

6 https://cloud.google.com/speech-to-text
spoken and parse it into a French sentence. Then, the French sentence can be translated to an English sentence, and the process can repeat. The system delights in that it progresses toward a more “French” sounding English phrase.

This small experiment parallels the surrealist linguistic work of Douglas Barbour in the 1980s with “homolinguistic translation” (Barbour and Scobie 1981). In that work, the author played with the sound of English phrases by turning them into something that the same sounds but wildly different meanings. For example, “The Pirates of Penzance” becomes “The Pirates of Pen’s Chance”.

We can very easily enable Automatic Translation Toward Meaninglessness within our Rosetta Code framework, simply by listening to text-to-speech synthesis using a loudspeaker instead of in-ear headphones. If both the microphone and the loudspeaker are on, the system will continuously feed on its own outputs. This situation can also be avoided by programmatically deactivating speech recognition during text-to-speech synthesis, or by using a microphone with an on-off switch that is triggered only when a person talks.

Methods for Multilingual AI Improv

We further combined the idea of multilingual improv with real-time translation, as we do in Rosetta Code, to implement a non-English version of AI improv comedy such as Improbotics and to enable an existing AI improv chatbot to work in multiple languages. Specifically, we have made two multi-lingual version of AI improv shows, one performed in Sweden (in Swedish and in English) and one in Belgium (in Flemish and in English).

Out of several options to perform the local language Improbotics show, the casts considered: a) fine-tuning the models into another language, b) completely retraining the model from scratch, or c) keeping the chatbot as is, but adding translation from and to a different language. This last option can either: 1) translate to the target language when pronounced in the earpiece and keep the interaction in English in the interface, or 2) translate from and to the language of the chatbot and thus allow the target language in the interface.

The first option, fine-tuning the GPT-2 model into another language, e.g. using Dutch corpora, assumes that the transformer model learned transferable language structures internally, giving an advantage when fine-tuning given a different language. While there are successful multilingual transformers, e.g. multilingual BERT (Devlin et al. 2018; Pires, Schlinger, and Garrette 2019), these were trained from scratch on multiple languages. A multilingual GPT-2 did not exist at the time of the production, so fine-tuning would have needed to use the English GPT-2. While using English tokenisation has performed decently for Dutch transformer models in classification tasks (Delobelle, Winters, and Berendt 2020), GPT-2’s English tokenizer, vocabulary and pre-training would have likely limited the linguistic correctness of generated Dutch sentences. The second option (retraining a GPT-2 model from scratch into the target language and then fine-tuning) was too costly for the production of the show (Synced 2019). We therefore chose the third option, i.e. to add a translation service, and thus treat the improvising AI like a Rosetta Code actor. The interface translates all human input from Dutch to English, and translates all the AI responses from English back into Dutch. Since the inputs are typed manually, and since Dutch-English translation is of high quality, the overall performance of the AI improv does not suffer. Using the translation service also opens up interesting routes for the future of Rosetta code. For instance, we are developing structures where the chatbots whisper generated responses to different performers in languages that they may or may not speak.

Rosetta Code on Stage

Rosetta Code was designed in 2018 and performed as a full show twice in November 2019 at the Voila! Europe Theatre Festival, “multilingual festival, often programming performances that use 2-3 languages in the same show” (Deyzac and Tasker 2020) at the Rich Mix theatre.7 The framing of the show was detailed in the shows’ description in the festival program, and on-stage at the beginning of the performance, making clear to the audience that the computer assists the creativity of the improvisers using automatic translation (Colton, Charnley, and Pease 2011). Covering 9 languages (Arabic, Dutch, English, French, German, Italian, Polish, Norwegian, and Swedish), it was presented as a multi-lingual Turing Test (Turing 1950) where the challenge for the audience was “to decipher who is human and who is a robot”8.

The Rosetta Code system was situated in the theatre setting. The show itself contextualized the technical details of the show. The show received positive response from performers, audience members, and critics. For instance, one quote from a performing musician reinforces the novelty and innovation of the show:

If you get a chance to see these guys, and it’s cross country, then see them. Polyglots improvising cross languages with other polyglots/native speakers. Improvisers improvising with AI. They will blow your mind. I have been lucky to improv with them the last few nights.9

The festival organisers observed the comedic potential of translation and miscommunication:

Rosetta Code is a fascinating experiment in how we create meaning. There were times when the audience had more understanding than the performers, as we could see the projected translations as well as gestures and body language. This imbalance of understanding can be a rich source of comedy.

The show was previewed and reviewed by multiple theatre critics and experts in innovative technical improvisation.7 Select quotes from reviews of the show include reference how the technology initially is intimidating but ultimately augments the humans. They also acknowledge that the system is not perfect, but rather, that there is beauty in the mistakes that are made.

---

7 http://j.mp/rosetta-code-supplementary
8 https://richmix.org.uk/events/rosetta-code/
what started as a potentially daunting evening . . .
decided upon being one of light entertainment, sprinkled
with some good humour and some sophisticated
technology used to a less sophisticated, but definitely
charming, purpose.

Observers also noted that the AI tools that underlay the
show build iteratively over the course of the performance.

The AI is first featured in the background of things —
providing translations or coming up with full dialogue
sentences delivered to the actors via a headset, which
they then act out on stage — and later at the forefront,
with the above mentioned, absolutely non-scary robot
taking center stage, teamed up with a human actor for
some one-on-one improvisation.

An edited video, based on the one-hour performance of
Rosetta Code, has been uploaded to the internet and can be
publicly accessed; a copy is provided for review.7

Finally, the Rosetta Code technology has also been de-
ployed in two local language AI Improv shows in Sweden
and in Flanders [references redacted for blind peer-review].

Historical Context, Related Work, Discussion
Rosetta Code is a first attempt at building augmentative tools
for performing translation-based theatrical improvisation in
any language. There are many challenges (some unforeseen)
to building and deploying such a system. In this section we
discuss challenges associated with improv in any language,
explore the cultural importance of multi-lingual artistic per-
f ormance, and provide several directions for future work.

Historical Context
Communication with and the understanding of other humans
are fundamental properties of the shared human condition
which date back to the origin of civilization. The chal-


7

7

5


... it is argued that the translation strategies adopted
by the media contribute to actively shape international
relations, and that translation activities deserve to be
attentively taken into consideration by policy and opin-
ion makers, as well as by the general public (Zanettin
2016).

Rosetta Code provides a platform to explore the power
of translation in meaning making between collaborating
humans. We acknowledge that there is much to be un-
derstood about others through interactions that are non-
language based. Dialogue between individuals, and per-
fomers on stage, can happen through many means, includ-
ing but not limited to vocalized language. They might use
body language, size, shape, and speed to construe meaning
to each other. We note that these channels of communica-
tion are complimentary and can be used in concert to build
shared understanding between people.

Improv, Thinking, and Language

Verbal and Non-verbal Improvisation
Improvisational theatre is theatre that is created and performed at the same
time. It can be seen as a constrained human interaction
game and has been qualified as “real-time dynamical prob-
lem solving” (Magerko et al. 2009; Johnson-Laird 2002)
in the settings of both jazz music and theatre. Improv requires
performers to exhibit, among others, acute listening to both
verbal and non-verbal suggestions coming from the other
improvisers, short- and long-term memory of narrative and
character elements, and practised storytelling skills (John-
stone 1979). One could categorise the behaviour of impro-
visers using the Dual Process psychological theory (Wason
and Evans 1974). That theory distinguishes “system 1” cog-
nitive processing corresponding to fast, intuitive, instinctive,
and emotional reactions—which can be honed using the-
atrical actor training practice (Benedetti 1999) to be able
to react truthfully in the moment (Meisner and Longwell
2012)—and “system 2” reasoning which is slower, more de-
liberative, logical and, in the case of improv, often more ver-
bal (Evans 1984; Kahneman 2003).

The Languages of Improv
Improvisation is performed in many
language differences around the world.9 Many times, impro-
visors share the stage with others who do not speak the same
language in the first language. In these situations, the performers must find a
level of scenic understanding using more than linguistics and
meta-pragmatic dialogue (Sawyer and Sawyer 2003). They
must respond to experience with intuition (Spolin 1963), and
make inferences about meaning. Sometimes the performers
resort to using gibberish, finding that it might be easier to
use a language that none of them, nor the audience under-
stand (Johnstone 2014).

Misunderstanding in Improv
In improvisation, there is an inherent tension between understanding and
misunderstanding. This is based on the nature of the artform. It is
an artform where choices are simultaneously made and under-
stood in the same moment. The audience is also experi-
encing these choices in the same moment that the collabo-
rating improvisors on stage hear them. Moment-to-moment

understanding is critical to the progression of an improvised performance.

**Improvisation for Language Learning**

Improvisation has been used as a form of language practice and learning. Theatrical improvisation promotes a special kind of verbal flow that may be particularly well suited to language learning (Egbert 2003). By engaging in role-playing activities, improvised with a variety of scene partners, language learners can engage in situated experiential learning (Butt 1998).

**Augmentative AI Language Technology**

Modern speech recognition, machine translation and dialogue systems, three popular fields of research in artificial intelligence (AI), share the same underlying mechanism: statistical language models trained on large corpora of text (Brants et al. 2007). Modern language models are implemented as neural networks and they estimate the likelihood of the next word or character token given some context about previous tokens (Bengio et al. 2003). Relatively similar neural networks models can be used for translation (i.e. generating a sentence in language B that corresponds to a context in language A), speech recognition (i.e. generating a sentence of symbols that corresponds to a sequences of acoustic phonemes), and in chatbots (i.e. generating sentences of dialogue likely to follow a given conversation context). Some examples of such modern text generation models include GPT-2 (Radford et al. 2019), Turing-NLP (Rosset 2020) and Meena (Adiwardana et al. 2020). These models cannot ground language understanding in the human sense. They manage to recognise speech, translate sentences with near human-level accuracy, generate plausible responses, and solve language comprehension tasks all owing to the large amounts of training data and a well-defined training objective.

These limitations can be linked to classic theatre theory and actor preparation. The seminal Russian theatre director and theorist Stanislavski explained that “the mistake most actors make is that they think about the result instead of the action that must prepare it” (Stanislavski 1989). That is, most actors understand what to say next, but not why they might say such a thing, or how it might be organized in a deliberative and impactful way. This mirrors the capacity of the computational language models, which know the likelihood of what might come next, but not why or how it should follow. Neural models have a grammar, as Chomsky (2002) would describe it: “a device of some sort for producing the sentences of the language under analysis” but not a rationale as to why such a device is meaningful in context. Such models can generate sentences which are grammatically correct and semantically nonsensical (Chomsky 2002).

These language models serve as building blocks for augmentative theatre technology, allowing performers to bring AI onto the stage. Unfortunately, many of these models, in particular, dialogue systems, are trained on large data-sets of strictly English text, which limits the applicability of such technology for non-English improv. We demonstrated several ways of overcoming these obstacles by combining chatbots with translation for multilingual AI improv shows.

**Conclusion**

In this paper we have presented Rosetta Code, a technology-based show structure that allows for improvised theatre in any language. We described the technical details underlying the system, and the theatrical context within which the system is situated. Rosetta Code is a show about language understanding via communication, that uses and celebrates the tools developed specifically to enable communication, namely speech recognition, machine translation and text-to-speech. We provide results and analysis of the first 3 shows covering 9 languages, as well as evaluative feedback from performers, producers, audiences and critics. Future work will involve systematic quantitative evaluation, enabled both by online performances and by sharing the tools with more improv troupes. Our framework allows for performers and audiences around the world to enjoy improvised theatre in any language, and we hope that these tools and techniques will empower and augment the art form. We will also explore incorporating language technologies with computationally creative systems for video and music generation.

**Acknowledgements**

The authors would like to thank Sharlit Deyzac and Amy Clare Tasker from Voila Europe Festival of European Theatre, as well as Mats Eldöen, Alice Major, Adam Meggido, Julian Faid for their help during the design, development, and deployment of Rosetta Code. Thomas Winters is a fellow of the Research Foundation-Flanders (FWO-Vlaanderen). The performers and creative cast in the November 2019 performances of Rosetta Code included: Boyd Branch, Harry Turnbull, Jenny Elving, Kieran Stalard, Kory Mathewson, Marouen Mralhi, Paul Little, Piotr Mirowski, Rhiannon Jenkins, Roel Fox and Sarah Davies.

**References**


Creating Six-word Stories via Inferred Linguistic and Semantic Formats

Brad Spendlove and Dan Ventura
Computer Science Department
Brigham Young University
Provo, UT 84602 USA
brad.spendlove@byu.edu, ventura@cs.byu.edu

Abstract
High-quality human-created artifacts are distinguished by cohesion and semantic richness that can be difficult for computational systems to emulate effectively. Certain classes of artifacts feature relationships between their constituent elements (e.g., words in a story) that naturally form a hierarchical structure that underpins the artifact’s meaning. We formalize a tractable method for programmatically extracting such artifacts’ structures framed in hierarchical Bayesian program learning (HBPL) and present HIEROS, a creative system that uses linguistic and semantic structures—collectively called formats—extracted from human-written six-word stories to guide the creation of novel stories. We describe how HIEROS infers formats from exemplar stories, how those formats inform its selection of words when writing novel stories, and how the system evaluates its stories to search for high-quality output. We present and evaluate stories HIEROS has written and discuss how our format model enhances the system’s creativity.

Introduction
Certain classes of creative artifacts are built atop a latent hierarchical structure in which the artifact’s constituent parts influence one another to various degrees. Examples include scenes in a movie, chapters in a novel, or notes in a song. We will refer to this class of artifacts as hierarchically structured artifacts that consist of a sequence of elements, with the relationships between those elements forming the edges of an acyclic hierarchy graph.

In addition to informing the artifact’s form and meaning, this underlying hierarchical structure can yield insight into the processes by which the artifact was created. The ordering of the elements in a given hierarchy mirrors, to some extent, the creative process of a human creator. Altering the relative importance of these concepts focuses the creative process in different ways. For example, a songwriter may start with a melody then build harmonies or start with a certain chord progression and write a fitting melody.

We consider English-language stories as an example class of hierarchically structured artifacts whose constituent elements are words. In particular, we will examine six-word stories, a genre of microfiction that attempts to tell a meaningful or interesting story in just six words. The brief nature of six-word stories presents an interesting creative challenge and facilitates modeling and discussion. Although stories are the focus of this paper, any artifact that can be described in terms of a hierarchy of relationships between its elements can be analyzed using the approach presented here.

The words in a story artifact have hierarchical relationships with one another. The words are related to or influence each other both linguistically, such as in tense agreement between nouns and verbs, and semantically, such as in whether a certain pairing of words makes logical sense or conveys an intended action or mood. Not all of these relationships carry the same importance to the resulting story, however. The set of relationships between words in an artifact can be arranged into a hierarchy of dependencies ordered by importance. The exact meaning of “importance” is left purposefully vague here. Many valid hierarchical constructions exist for any artifact which may assign different importance to inter-word relationships or feature different relationships altogether.

For example, we can consider an underlying hierarchical structure to the story “the dog runs” in which the second word in the story is the most important and informs the first and third words in different ways. The relationship between the second word and the first is likely not as strong or important as the relationship between the second and the third. Furthermore, there are many ways to interpret the relationship between the second and third words, such as that they are merely a noun and a verb, that they are a subject and an action that that subject could take, or that they are a subject and action that specifically convey energy or urgency.

Note that the ordering of the hierarchy may not correlate to the sequential order in which the words are arranged in the artifact. Words form the artifact, and the relationships between them form the hierarchy. Other words could fit those same relationships, resulting in a different story but adhering to the same hierarchy. The story “a whale breaches” could be said to have the same underlying hierarchical structure as “the dog runs”, such as the structure of a story featuring an animal subject, an article preceding that subject, and a verb that reflects the subject’s energetic movement.

As language and its meaning are subjective, there may exist many possible hierarchical structures underlying any given artifact. The more a given hierarchy captures the interesting properties of the artifact and its meaning, the more
useful that hierarchy is. Another example story “the dog cowers” could be considered to have the same relationship between ‘dog’ and ‘cowers’ as the previous example did between ‘dog’ and ‘run’, but there may be an important semantic difference between ‘runs’ and ‘cowers’ that a more useful hierarchy could reflect.

If a computational system can model an artifact’s hierarchical structure, it can, by extension, model the whole suite of artifacts that fit in that same structure. A method for programmatically extracting useful hierarchies from creative artifacts would allow a computational system to model the relationships between an artifact’s elements, thereby identifying the key components from which the artifact’s meaning and quality are derived.

In this paper, we present HIEROS a computationally creative system for writing six-word stories based on a hierarchical model of story structure. HIEROS infers the hierarchical structures of human-written exemplar stories by constructing formats that are used to guide its creation of novel stories. We present the underlying model of story hierarchy that HIEROS implements, describe the system in detail, and analyze its results.

Related Work

While our approach to inferring formats for story generation is novel, previous creative writing systems also involve elements of constrained generation and learning from human-written artifacts. The models underlying many of these systems can be usefully viewed as hierarchical (or at least relational), allowing for more direct comparison to our system. In this section, we refer to several such systems and contrast them to the novel techniques used in HIEROS.

The storytelling systems MEXICA (Pérez y Pérez and Sharples 2001), STellA (León and Gervás 2014), and Fabulist (Riedl and Young 2006) all seek to guide the creation of story artifacts by imposing useful restrictions on the space of possible story events. These restrictions represent a hierarchy of probability distributions conditioned on previous events. HIEROS explicitly models such hierarchies and infers them from human-written exemplars instead of drawing from hard-coded concepts.

Both Colton et al. (2012) and Toivanen et al. (2012) present creative poetry-writing systems that model an exemplar’s syntactic structure as a template for creating novel artifacts. The former system constrains the populating of the template with rules regarding the relationships of possible words with one another, while the latter models semantic relationships in a separate corpus to guide creation. These systems use fixed sets of constraints that are applied to all exemplar-generated poems, while HIEROS infers semantic constraint information from exemplars which are directly incorporated into its formats.

MICROS (Spendlove, Zabriskie, and Ventura 2018) presents a prototypical exploration of latent hierarchical structure in its automatic creation of six-word stories. The MICROS system creates story artifacts that are built upon an underlying structure, demonstrating the power of this model in guiding the creative process. MICROS’ single, static story structure allows it to draw on powerful semantic knowledge bases but also heavily restricts its output variety. HIEROS shares MICROS’ underlying model but improves upon MICROS by removing its fixed format restriction. Removing this restriction requires a more generalized method of generating and scoring stories and a method for automatically inferring templates from exemplars. The result we present here is a system that can generate stories according to a much broader set of formats than MICROS can.

Li, Wu, and Lan found that augmenting existing hidden variable models with syntactic and semantic structures improved their performance at machine comprehension tasks, demonstrating how modeling such structures is useful outside the domain of creativity (2018). We argue that our model is useful for any domain which concerns hierarchically structured artifacts, not just story artifacts. Any domain that permits viewing its artifacts as composed of elements and the relationships between those elements could be framed in this model for use in tasks not limited to generation.

Frame semantics (Fillmore and Baker 2001) is a linguistic theory that views a concept as a “frame” consisting of an arrangement of elements that comprise that concept, with each element representing a set of words that fulfill a specified role in that concept. Our model’s formats bear some similarity to semantic frames in that both structures place restrictions on which words are acceptable to communicate a story or a concept, respectively. One implementation of frame semantics, FrameNet (Fillmore, Wooters, and Baker 2001), consists of a database of hand-annotated sentences and their resulting semantic frames. Our system seeks instead to infer formats automatically from human-written exemplars, allowing it to draw on a broader set of formats for generation.

Modeling Latent Hierarchical Structure

Under HBPL

In this section, we develop a formal model for the general hierarchical structure of six-word stories. Similar models may be constructed for all types of hierarchically structured artifacts, with their constituent elements replacing words as the atomic units whose relationships the hierarchy captures.

Although the meaningful relationships between words in a story artifact may involve complex, recursive connections, in our model we simplify this hierarchy to a directed acyclic graph with at most one edge between any two elements.

If a directed edge exists from element p to element c, we refer to p as c’s parent and c as p’s child. We represent an artifact’s hierarchy graph as a joint probability distribution over all possible words factorized into a series of subdistributions in which each word is conditioned only on its parent word in the hierarchy. Each subdistribution thus captures a relationship in the hierarchy by assigning high probabilities to words that follow from the parent word according to that relationship.

Thus, for the story “the dog runs” in which the hierarchy graph includes an edge (i.e. relationship) from the second word to the third word, ‘runs’ can be thought of as being drawn from a distribution of possible words conditioned on
the parent word ‘dog’. If the relationship represented by this distribution is that the third word is an action that the second word could take, then other high-probability words in the distribution could be ‘barks’ or ‘sits’. However, if the relationship is instead more specifically characterized as an energetic movement, the high-probability words could include ‘sprints’ or ‘flies’. Of course, these are only two of the many possible relationships that could describe the words’ meanings in relation to one another.

Hierarchical Bayesian program learning (HBPL) (Lake, Salakhutdinov, and Tenenbaum 2015) provides a useful framework for modeling story artifacts as factorized joint probability distributions. Let a story \( S = w_1 \ldots w_n \) be a sequence of \( n \) words \( w_i \) with \( w_i \in \mathcal{W} \), the set of all possible words. Then a probabilistic approach to the problem of story creation imposes a joint distribution \( p(S) = p(w_1, w_2, \ldots, w_n) \) over the set \( \mathcal{S} \) of all possible stories. Thus, creating a story means simply sampling from \( p(S) \).

Of course, for stories of any length, this distribution is likely to be intractable to compute, and thus typically some simplifying assumptions are made that allow the joint distribution to be factored in some way.

HBPL suggests that there are domain-specific factorizations that both simplify the computational demands of this generational approach and that exhibit explanatory power as well. Such factorization mirrors the human creative process both by providing focus and by reducing the large space of artifact creation to a series of smaller, tractable creative decisions.

For the domain of \( n \)-word microfiction stories, the most complete factorization comes from an application of the chain rule:

\[
p(S) = p(w_1, w_2, \ldots, w_n) = p(w_1)p(w_2|w_1) \ldots p(w_n|w_{n-1})
\]

where \( i_j \in [1, n] \) and \( i_j \neq i_k \) unless \( j = k \), so that this represents a general version of the chain rule that admits any possible permutation of word order dependency.

Thus, we have a formal model for hierarchy structure that allows any ordering of relationships between words and their corresponding positions in the final story. The factorization is hierarchical—each subdistribution beyond the first is conditioned on a word chosen from a subdistribution preceding it—and the formalism says nothing about how the probabilities for a given subdistribution are calculated, allowing for any and all interpretations of the relationships between the words in the story. If the structure and relationships imposed by a given factorization are “good”, the result should be “good” stories. Similarly, a good factorization should be more tractable to compute than the whole joint probability.

In what follows, we refer to a particular factorization of the joint as a story format that represents one possible hierarchical structure. The relationships described by the conditional subdistributions of a given format determine what the format represents. The trivial format is one in which each subdistribution assigns all probability to a single word; sampling from that joint will always yield the same story artifact. A better format could, for example, capture linguistic relationships, assigning high probability to all words of a given part of speech. A further improved format could represent the semantic relationships between words and assign high probabilities to words that combine to form meaningful stories.

Computationally modeled formats with tractable subdistributions would be powerful tools to aid machine comprehension and generation of creative artifacts. Such formats could be manually constructed, as in the MICROS system, but automatically inferring the underlying semantic format of a human-written exemplar allows the system more flexibility and breadth of results. Indeed, we argue that such a system exhibits more creativity.

We have developed a generalized six-word story writing system that leverages the power of this model to create stories whose underlying structures mirror the linguistic coherence and semantic richness typified by human-written stories.

HIEROS

To test the efficacy of our model we developed HIEROS, a creative system that writes six-word stories using the inferred formats of human-written stories to guide the linguistic and semantic choices it makes when selecting words to create a novel artifact. The system operates in three main steps: inferring formats from exemplar stories, generating new stories based on those formats, and evaluating those stories by assigning each a quality score. Generated and scored stories are then refined by repeatedly mutating high-scoring stories via a modified generation step.

Inferring Story Formats

In order to model the meaning and quality of human-written stories for use in novel artifact creation, HIEROS infers formats from a list of exemplar stories that the system takes as input. In particular, the inferred formats contain linguistic (part-of-speech) information as well as an approximation of the semantic relationships typified by the words in the exemplar. The system first constructs a hierarchy of parts of speech and dependency relationships then fills in semantic information pertaining to those relationships.

The structure of the format is constructed using the Stanford Parser (De Marneffe, MacCartney, and Manning 2006) which statistically parses the exemplar and extracts both the parts of speech of each word and the dependencies between the words, forming a parse tree of dependency relations. Such relations reflect directed linguistic links between words in the artifact.

Given a six-word story exemplar \( X = x_1, x_2, x_3, x_4, x_5, x_6 \) with \( x_1 \in \mathcal{W} \), the parser constructs a directed acyclic graph \( G = (V, E) \) with \( V = X \). Each dependency in the parse tree is represented by a directed edge \( (x_p, x_c) \in E \) where \( x_p \) is the parent word and \( x_c \) is the child word. One word \( x_r \) has no parent and serves as the root of the parse tree.

The dependencies represented by edges in the parse tree dictate the ordering of the words \( w_1 \) through \( w_6 \) in the format’s factorization of the joint \( p(S) \), such that
\[ p(S) = p(w_r) \prod_{E} p(w_c|w_p). \]

Finally, \( \psi(x_i) \), where \( \psi \) is a function that returns a word’s part of speech, is recorded for each \( i \in [1, 6] \), completing the linguistic hierarchy.

Automatically extracting semantic information from text is challenging. Although HIEROS’ scope is limited to identifying relationships between just two words at a time, its method for constructing \( p(S) \) must be able to model all possible semantic relationships between words that may exist in the formats inferred from exemplar inputs. To model such semantic relationships, HIEROS draws on the linguistic theory of distributional semantics, which hypothesizes that words that have similar meanings appear in similar contexts (Sahlgren 2008).

Specifically, HIEROS uses word2vec word embeddings (Mikolov et al. 2013) to model distributional semantic information. Being trained on a large corpus of data, in this case Wikipedia, the word embedding model represents each word’s aggregate context as a many-dimensional vector. Thus, all the words in the corpus are mapped into one vector space that represents their relative contexts and therefore relative meanings.

Hierarchical ordering of the words is a function of the semantic relationship between those two words. The system traverses each edge in the exemplar’s parse tree and calculates \( \psi(x_c) - \psi(x_p) \), where \( \psi \) is the word embedding function. It records the resulting semantic vector for use in identifying other words that share the semantic relationship that the edge represents. Thus, the system can be said to reverse engineer the semantic relationships between words in a human-written exemplar to use as a template for generating novel stories.

With these steps completed, HIEROS has constructed a format that represents the hierarchical ordering of the words in the exemplar, their parts of speech, and the semantic relationships between the dependent words. This corresponds to a factorization of \( p(S) \) in which each subdistribution is conditioned on the word that precedes it in the hierarchy. The part-of-speech and semantic vector data recorded for each subdistribution will be used at generation time to assign high probabilities to words that fulfill the linguistic and semantic restrictions corresponding to those values. The format informs how the subdistributions should be constructed; the generator constructs and samples from them to create a story.

Figure 1 shows an example story (1) that is parsed to form a hierarchy of dependencies (2). That hierarchy is converted into a format (3) which can be visualized as a graph in which each node records an index into the story and a part of speech, and each edge stores the semantic vector calculated from the corresponding words in the exemplar. Note that the exemplar words themselves do not form part of the format.

The exemplar’s parse tree dictates the format’s factorization of the joint, in this case:

\[ p(S) = p(w_1, w_2, w_3, w_4, w_5, w_6) = p(w_1)p(w_3|w_1)p(w_4|w_1)p(w_2|w_3)p(w_6|w_4)p(w_5|w_6). \]

Recall that for each subdistribution \( p(w_i|w_j) \)—corresponding to the edge \((x_j, x_i)\) in the hierarchy graph—the format stores an associated semantic vector \( \psi(x_j) - \psi(x_i) \) and for each \( w_i \) the format stores a part of speech \( \psi(x_i) \). These values are used to construct each subdistribution.

**Generation**

Our HBPL model of story creation views writing a six-word story as sampling from \( p(S) \). To accomplish this, the generator selects one inferred format at random, calculates the probability distributions for each factor of the joint represented by that format, and samples them to yield a six-word story.

**Sampling from Subdistributions** HIEROS begins generation by constructing the first subdistribution \( p(w_r) \), which represents the root of the format’s hierarchy, and sampling from it. Because no word precedes the root, there is no semantic relationship to consider when assigning probabilities, and the subdistribution is populated by common words that match the given part of speech. To identify these words, HIEROS employs a list of the most common English words from the Corpus of Contemporary American English (Davies 2010).

To populate the root subdistribution, all words from this list that match the appropriate part of speech are collected and sorted from most common to least. Then the top 10% of the words, rounded down, are discarded in order to filter out bland or overly common words. The remaining words are each assigned equal probability to form the first subdistribution, which HIEROS samples to select the root word.

The format’s hierarchy dictates an ordering for building the remaining subdistributions \( p(w_c|w_p) \) and sampling \( w_c \).
for \( c \in [1, 6] - r \). HIEROS constructs these subdistributions as follows.

The independence assumptions provided by our model’s application of the chain rule reduce the large space of possible words into the tractable problem of identifying words of a certain part of speech that fulfill a given semantic relationship with a preceding word. Recall that the inferred format contains for each subdistribution a part of speech \( \psi(x_c) \) and a vector \( \vec{v}(x_c) - \vec{v}(x_p) \) representing the semantic relationship that should exist between \( w_p \) and the words with non-zero probability in \( p(w_c | w_p) \).

The vector \( \vec{v} = \vec{v}(w_p) + \vec{v}(x_p) - \vec{v}(x_c) \) corresponds to a point in the word embedding space, such that the set

\[
Y = \{ y \in \mathcal{Y} | \alpha > \| \vec{v} - \vec{v}(y) \| \land \beta < \| \vec{v} - \vec{v}(y) \| \}
\]

contains words that are related to \( w_p \) according to the relationship dictated by the semantic vector. \( \alpha \) and \( \beta \) are constants that bound the minimum and maximum distances of a word’s embedding from \( \vec{v} \) for it to be considered related to \( w_p \) in this way. \( \beta > 0 \) prevents words that are too closely related to \( w_p \) from populating the subdistribution, as these words are likely bland or uninteresting.

HIEROS constructs \( Y' = \{ y' \in Y | \psi(y') = \psi(x_c) \} \), resulting in the final list of words to which equal probabilities will be assigned to populate \( p(w_c | w_p) \). HIEROS samples \( w_c \) from the resulting distribution.

Once all six subdistributions have been constructed and sampled, the system has sampled \( S = w_1, w_2, w_3, w_4, w_5, w_6 \) from \( p(S) \), completing one round of generation.

**Mutating via Resampling** As refining HIEROS’ generated stories involves mutating a generated story, the generator is designed to efficiently resample \( p(S) \) by selecting a word \( w_m \) and drawing a new word \( w'_m \neq w_m \) from the same subdistribution from which it was originally drawn.

Due to the hierarchical nature of the format, this reselection may necessitate cascading changes to other words in the story. If the selected word has dependencies below it in the format’s hierarchy, those subdistributions will be constructed anew, conditioned on the newly selected preceding word, and sampled to select new words. If dependencies are found below those words, those subdistributions will be re-constructed and resampled, and so on. Thus, by changing one word in the story, the mutation process may result in a new story that differs from the original by more than one word.

**Evaluation & Refinement**

HIEROS improves the quality of its stories by evaluating and refining them. It scores its generated stories and mutates them to observe whether resampled stories score higher, continuing until no higher scoring stories are generated.

**Scoring** Assigning scores to creative artifacts that reflect their quality is a key challenge for any computational system that relies on a generation-evaluation loop. HIEROS employs the same skip-thought scoring method as MICROS. Skip-thought vectors (Kiros et al. 2015) are similar to word2vec word embeddings in that they encode natural language strings as vectors in a high-dimensional semantic space, but whereas word2vec maps words to vectors, the skip-thoughts model maps sentences to vectors.

Skip-thought vectors can be used to score a story \( S_i \) by measuring its similarity to two high-quality stories \( S_p, S_h \) and its dissimilarity from a poor-quality story \( S_l \), where \( S_i \in \mathcal{S} \).

Let \( \bar{a}_1 = \tau(S_p) - \tau(S_h) \) and \( \bar{a}_2 = \tau(S_h) - \tau(S_l) \), where \( \tau \) is the story embedding function. Let \( \vec{n} = \bar{a}_1 \times \bar{a}_2 \), and let \( \phi_{\vec{n}} \) be the function that projects a vector onto \( \vec{n} \). Let \( \vec{s} = \tau(S_i) \). Then the score for \( S_i \) is \( \| \vec{s} - \phi_{\vec{n}}(\vec{s}) \| \), or the magnitude of \( \vec{s} \) when it is projected onto the plane defined by \( \bar{a}_1 \) and \( \bar{a}_2 \) whose origin is at \( \tau(S_h) \). Thus, poor-quality stories have vector representations that project closer to the origin, and the vectors for higher-quality stories are projected further from it.

Because the scoring plane is described by vectors from \( S_h \) to \( S_p \) and \( S_h \), we refer to the high-quality stories as “axes”.

Choosing which stories to use as axes is a critical consideration for this scoring method. HIEROS leverages its list of exemplar inputs to select high-quality axis stories for scoring.

Ideally, the two axes are distinct in order to capture as much information in the score as possible. Furthermore, the axes should ideally not include the exemplar story that inspired the generated story to be scored; otherwise high scores will be assigned to generated stories that are very similar to their exemplars. However, if the axes are too different from the story to be scored, the score will be less accurate. To balance these considerations, HIEROS prefers to select axes that are distinct from the exemplar story but that still share similar parts of speech with that story.

Let \( P(X) \) represent the concatenation of exemplar \( X \)’s parts of speech: \( P(X) = \psi(x_1) \| \psi(x_2) \| \ldots \| \psi(x_n) \). Let \( C = \{(X_i, X_j) | P(X_i) = P(X_j)\} \). For each exemplar \( X_i \), HIEROS chooses at random \( X_j \) such that \( (X_i, X_j) \in C \), with \( i \neq j \) if possible. \( X_i \) and \( X_j \) then serve as scoring axes \( S_p \) and \( S_h \) for all stories generated with the format inferred from \( X_i \). If \( X_i \neq X_j \), we refer to these axes as “diverse”, otherwise we refer to them as “single” axes.

We experimented with using different axis configurations to score HIEROS’ stories. We discuss the trade-offs between different axis selection methods in a later section but note here that we chose to use diverse axes.

**Refinement Process** HIEROS’ refiner organizes generated stories by root word, maintaining a priority queue containing the highest scoring stories with that root that it has generated so far. These queues are initially seeded by generating one story for each possible root word (i.e., each word with non-zero probability in the format’s first subdistribution) instead of selecting a root randomly. With the priority queues thus populated, refinement then proceeds in steps.

At each refinement step, the highest scoring story in each queue is popped and mutated to generate a specified number of children. Each of those children is scored and placed into the priority queue corresponding to its root (which may differ from its parent’s due to mutation). Each queue also
remembers the highest scoring stories it has seen across all refinement steps. After each step, if its highest-scoring story has not changed, a counter reflecting that queue’s “staleness” is incremented. Once that counter reaches a specified number, the queue is considered stale and no more stories are popped from it. When all queues are stale the refinement process terminates, and the two highest-scoring stories from each queue are collected to form the output of the refiner.

Due to the limitations of skip-thought scoring, some stories receive higher scores without being of higher general quality than other stories. This bias appears to be linked to certain words in a story. By maintaining several queues and taking the top stories from each, the system avoids biasing its output toward many similar stories that feature such words.

After refinement, the final step in generating stories for a given format is to capitalize and punctuate them according to the format’s exemplar. To take advantage of the variety of formats that may exist among the exemplar stories, HIEROS repeats this same generation and refinement process for several randomly selected formats. After refined stories have been generated for each format, the combined results are sorted by score, and a subset of the highest scoring stories is returned as the creative output of the system.

This subset includes a number of high-scoring generated stories, excluding a number of the highest-scoring of those stories, with both the specific numbers of inclusion and exclusion specified by parameters. Due to the scorer’s tendency to assign high scores to stories that more closely resemble the exemplar, the most interesting potential output may not include those stories with the highest scores.

Results

HIEROS’ results demonstrate that our formulation of story formats is a useful model to guide the automatic creation of six-word stories. Its ability to model the underlying structure of human-written stories results in broad variety among its created artifacts.

Coherence and impact are two complementary qualities by which six-word stories may be judged. Coherence refers to whether a story is understandable and makes sense, while impact refers to whether a story succeeds in eliciting an emotional reaction from the reader. These qualities parallel the latent structures that our system attempts to infer from exemplar stories, namely linguistic structure and semantic structure. A format that accurately models the former should generate coherent stories, while a format that models the latter has a higher chance of generating impactful ones.

Input & Parameters

We scraped exemplar six-word stories from Reddit\textsuperscript{1} and Twitter\textsuperscript{2}. We removed any stories that featured nonstandard spelling or symbols as well as others we deemed unsuitable for format inference, including stories that featured acronyms or pop culture references. This left us with 1,481 exemplars from which HIEROS inferred formats. HIEROS selects formats that have diverse scoring axes as candidates for generation first, falling back to those with single axes if it finds an insufficient number that are diverse. 261 of the inferred formats could be clustered by part of speech with one or more other formats, giving them two distinct scoring axes. These formats were used for story creation.

The parameters HIEROS used for story creation are as follows. Four children were created every time a story was popped from the top of its priority queue in the refiner, and the queue for a given root was considered stale if three rounds had passed without it seeing a new highest scoring story. Each time HIEROS ran, it generated stories for 30 formats chosen at random and selected as its output portion the 85th to 90th percentile of highest scoring stories. This resulted in the system outputting approximately ten stories each execution.

Survey

We conducted a survey to evaluate HIEROS’ results and the efficacy of its scorer using the output of ten executions— for a total of 100 stories—plus the 15 lowest scoring stories that the refiner saw over one execution. The latter group of stories would never be output from the system, but we included it to serve as an indication of how well the refiner is able to distinguish low-quality stories.

The survey briefly introduced six-word stories as a type of creative artifact, defined coherence and impact, and presented 15 randomly chosen stories for the respondent to evaluate. For each story, we asked the respondent to rate the degree to which they agreed with the following two statements on a seven-point Likert scale: “This story is coherent (understandable, correct English),” and “This story is impactful (meaningful, funny, sad, etc.),”.

We distributed the survey via social media and received 124 responses, including partial responses in which fewer than 15 stories were rated. We did not collect demographic data as part of the survey, but the majority of the audience to which the survey was presented were native English speakers. Each story was rated by an average of 14 respondents.

The results of the survey demonstrate that a handful of HIEROS’ stories achieve coherence, such as “‘To him, ‘endlessly’ meant ‘twenty decades.’” and “‘Joy is a path to childhood.’,” however the majority do not. Very few of its stories are both coherent and impactful, however some do manage to achieve this more elusive effect, such as “Diamond ring. Glassy diamond. Costliest engagement.”. Interestingly, while most stories were rated as having lower impact than coherence, some were found to be impactful despite being somewhat incoherent. One such example, “Find, yours cowardice is his strength.”, seems to evoke a response through the emotionally charged juxtaposition of “[your] cowardice” and “his strength” despite not telling an understandable story.

We can characterize the accuracy of the skip-thought scorer by comparing the ratings respondents gave to the top 15 highest skip-thought-scored stories compared to the 15 lowest scored stories. Figure 2 shows boxplots of the ag-

\textsuperscript{1}https://www.reddit.com/r/sixwordstories/top/?t=all
\textsuperscript{2}https://twitter.com/sixwordstories
\textsuperscript{3}https://twitter.com/ernest6words
Aggregates coherence and impact ratings of these two groups of stories. The respondents’ ratings of the top 15 stories’ coherence were significantly higher than their ratings for the bottom 15, $t(208) = 3.01, p = 0.003$. The respondents’ impact ratings were also significantly higher for the top 15 stories than for the bottom 15, $t(208) = 4.30, p < 0.001$, using an alpha of 0.01 for both statistical tests.

This demonstrates that the skip-thought scorer successfully assigns lower scores to low-quality stories. However, the scorer also assigns high scores to many stories to which respondents assign low ratings. Thus the scorer, and by extension the refiner, is limited in its ability to truly distinguish high-quality six-word stories.

**Discussion**

HIEROS’ results show that the system, while imperfect, is capable of writing interesting stories. Furthermore, because the system takes as input any set of six-word stories, its output is as varied as its input. This is demonstrated by HIEROS-created stories such as “No-one persists sympathetic. Dislike is unsympathetic.”, “His dirtiest bush is seldom artificial.”, “Adverse anger yells. Undesirable euphoria grieves.”, “A creamy maroon color turned currant.”, and “Disgust fixes each tenderness within misadventure.”. This ability to create a variety of occasionally-interesting stories indicates that HIEROS’ inferred formats do capture some degree of the exemplars’ structure and semantic meaning.

HIEROS’ refiner uses its skip-thought scorer to explore the myriad of stories represented by $p(S)$. The interplay between the quality of stories that are assigned high probabilities in that joint distribution and the stories to which the scorer assigns high scores determines the quality of the system’s final output. Axis selection is critical to the accuracy of the skip-thought scorer. We experimented with two main approaches to selecting axes: choosing $X_1$ at random such that $(X_i, X_j) \in C \land i \neq j$ and choosing $X_k$, $X_{k'}$ at random such that $(X_i, X_j) \in C \land (X_{i}, X_{k'}) \in C \land i \neq j \neq k \neq i$.

When the exemplar $X_1$ is used as a scoring axis, the scorer assigns high scores to stories that are more coherent but also more similar to the exemplar. When the exemplar is not included among the axes, we observe that high-scoring stories are more varied but of lower quality overall. In order for the score to better reflect story quality, we decided that it was permissible to bias the scorer somewhat toward stories that were similar to the exemplar.

To evaluate whether the skip-thought scoring method achieved its goal of reflecting story quality, we performed an experiment in which we scored HIEROS-generated stories as progressively more of their words were replaced with random words. If the scorer is able to accurately measure quality, then a story should score lower the more random words are present in it. Figure 3 shows the results of conducting this experiment on four randomly selected HIEROS-generated stories. These results demonstrate a general trend that less random stories correlate to higher scores.

Despite this demonstration that HIEROS’ scoring method can distinguish between random and non-random stories to some degree, it is clear that the scorer is primarily to blame for the low quality of the system’s stories overall. As evidenced by the best of its output, HIEROS’ generation and mutation systems are capable of writing quality stories. However, the scorer is not able to consistently identify those stories. A more accurate scorer would be more effective in directing HIEROS’ story generation and would improve the system’s results.

Similarly, the word2vec word embedding model that HIEROS uses could be improved or replaced in order to improve the quality of the system’s output. HIEROS’ word2vec model was trained on a Wikipedia corpus that, while large, does not include more poetic contexts for the words it contains, restricting the breadth of the model’s rep-
resentations of those words meanings. Training word2vec on a corpus of poetry or other literary works is thus a potential avenue for improving HIEROS. Alternatively, replacing word2vec with an improved model of semantic meaning could help prevent HIEROS' word selection from tending toward synonyms of exemplar words.

Finally, we note that the quality of a HIEROS-generated story is unlikely to surpass that of its exemplar story. Although the exemplar stories scraped from Reddit and Twitter represent relatively high-quality amateur stories, using exemplars written by more skilled poets would raise the ceiling on the quality of HIEROS’ stories.

Conclusion

Framing a creative artifact’s structure as a hierarchical factorization of a joint probability over potential elements allows a computational system to place useful restrictions on the elements it selects to generate novel artifacts. This model provides a tractably computable means for modeling the cohesion and richness of human-created artifacts, which is useful for imparting quality and generalizability to a creative system.

We have presented a model of hierarchical artifact structure, examined how it can be applied to the domain of six-word stories, and demonstrated a method for inferring structure from exemplar stories. Our creative system HIEROS produces a wide variety of output using this method, some of which achieves impactfulness, providing an argument for the utility of this model as a guide for creative generation.

References


Toward Automated Quest Generation in Text-Adventure Games

Prithviraj Ammanabrolu  William Broniec  Alex Mueller  Jeremy Paul  Mark O. Riedl
School of Interactive Computing
Georgia Institute of Technology
Atlanta, GA 30332 USA
raj.ammanabrolu@gatech.edu

Abstract

Interactive fictions, or text-adventures, are games in which a player interacts with a world entirely through textual descriptions and text actions. These games are typically structured as puzzles or quests wherein the player must execute certain actions in a certain order to succeed. In this paper, we consider the problem of procedurally generating a quest, defined as a series of actions required to progress towards a goal, in a text-adventure game. Quest generation in text environments is challenging because they must be semantically coherent. We present and evaluate two quest generation techniques: (1) a Markov model, and (2) a neural generative model. We specifically look at generating quests about cooking and train our models on recipe data. We evaluate our techniques with human participant studies looking at perceived creativity and coherence.

Introduction

Natural language can be used to express creativity in the form of narrative. Prior research has shown that narrative is used in everything from environmental understanding (Bruner 1991) to developing language (Johnston 2008). Given this wide ranging impact, using narrative in language to help us understand human perceptions of creativity and what it takes to replicate this through computational models is natural. Text-adventure games, or interactive fiction, in which a player interacts with a world entirely through text, provide us with a platform on which to explore ideas on creativity in language. These games are usually structured as puzzles or quests in which a player must complete a sequence of actions in order to succeed. Text games allow us to factorize the problem of creative language generation and focus on developing more fine-grained, data-driven models.

Automated generation of text-adventure games can broadly be split into two considerations: (1) the structure of the world, including the layout of rooms, textual description of rooms, objects, and non-player characters; and (2) the quest, consisting of the partial ordering of activities that the player must engage in to make progress toward the end of the game. In this work, we focus on methods of automatically generating such a quest and how it can be used to better understand narrative intelligence, specifically looking at perceived creativity and coherence. Quest generation requires narrative intelligence as a quest must maintain coherence throughout and progress toward a goal. Maintaining quest coherence also means following the constraints of the given game world. The quest has to fit within the confines of the world in terms of both genre and given affordances—e.g. using magic in a fantasy world. This is further complicated in the case of a text-adventure as a consequence of all interactions being in natural language—the potential output space is combinatorial in size. Because the player “sees” and “acts” entirely through text, any quest generation system must also take into account the lack of visual information and generate sufficiently descriptive text accordingly.

There are multiple variables that could potentially affect a player’s perception of creativity in a text-adventure game such as the vocabulary used, the structure of the world, stylistic variations in writing, etc. To conduct controlled studies, we use the TextWorld framework (Côté et al. 2018) which lets us generate text-adventure game worlds based on a grammar, allowing us to focus on novel quest generation algorithms. It lets us fix variables concerned with game world and logic generation and focus only on the generation of quests within this world. We use this framework’s “home” theme—providing us with a textual simulation of a house—and restrict the types of quests that can be generated to those involving the completion of a cooking recipe. We then attempt to learn how to generate a quest to complete a recipe—as well as how to create the recipe itself—using a large scale knowledge base of recipes. In these quests, players are provided with a list of ingredients and their locations, and they have to navigate the environment to find and prepare those ingredients to complete the given recipe. For example, given a recipe to make peanut butter cookies the quest would first tell the player to find eggs, peanut butter, flour, and baking soda. The player would then have to figure out that the first ingredient is in the fridge while the others are in the pantry and prepare each item accordingly. Generating this sort of quest requires knowledge of the ingredients, how they fit together, and how those ingredients interact with the environment.

The contribution of this work is thus twofold. We first detail a framework, and variations thereof, that can learn to generate creative quests in a text-adventure game. This framework includes two quest generation models using Markov chains as well as a neural language model. It also
uses a semantically grounded knowledge graph to improve overall quest coherence. Our second contribution provides human subject evaluations that give us insight into how each variation of this framework affects human perception of creativity and coherence in such games.

Related Work

Although there has been much work recently on text-adventure gameplay (Bordes et al. 2010; He et al. 2016; Narasimhan, Kulkarni, and Barzilay 2015; Fulda et al. 2017; Yang et al. 2018; Haroush et al. 2018; Côté et al. 2018; Tao et al. 2018; Ammanabrolu and Riedl 2019a; 2019b; Hausknecht et al. 2019a; 2019b; Ammanabrolu and Hausknecht 2020), these works focus on creating agents that can play a given game as opposed to being able to automatically generate content for them.

Outside of this, there has been some work on learning to create content in the context of interactive narrative. These systems mainly work to overcome a significant bottleneck in the form of the human authoring required to create such works. Permar and Magerko (2013) present a method of generating cognitive scripts required for freeform activities in the form of pretend play. Specifically, they use interactive narrative—a form of pretend play that requires a high level of improvisation and creativity and uses cognitive scripts acquired from multiple experience sources. They take existing cognitive scripts and blend them in the vein of more traditional conceptual blending (Veale, O’donoghue, and Keane 2000; Zoook, Magerko, and Riedl 2011) to create new blended scripts. Closely related is Magerko et al. (2014) who present a Co-Creative Cognitive Architecture (CoCoA), detailing the set of components that support the design of co-creative agents in the context of interactive narrative. These methods all follow singular cognitive models that do not learn to generate content automatically.

Li et al. (2012) present Scheherazade, a system which learns a plot graph based on stories written by crowd sourcing the task of writing short stories through Amazon Mechanical Turk. This plot graph contains details relevant for the coherence of the story and includes: plot events, temporal precedence, and mutual exclusion relations. The generated narrative contains events that can be executed from this plot graph by both players and non-player characters. Guzdial et al. (2015) introduce Scheherazade-IF, a system that learns to generate choose-your-own-adventure style interactive fictions in which the player chooses from prescribed options. More recently, Martin et al. (2017) introduce a pipeline systems for improvisational storytelling agents capable of collaboratively creating stories. These agents first focus on creating a plot for the story and then expand that plot into natural language sentences.

Giannatos et al. (2011) use genetic algorithms to create new story plot points for an existing game of interactive fiction using an encoding known as a precedence-constraint graph. This graph gives the system information regarding the ordering of events that must happen in the game in order to advance. They demonstrate the workings of their system by generating additional content for the popular interactive fiction game Anchorhead, and show that this can be integrated into the original game. This work, however, is offline and relies on existing interactive fiction games and having knowledge of the precedence-constraint graph for this existing game.

The Game Forge system (Hartsook et al. 2011) also uses genetic algorithms to generate a game world and plot line for related type of game, a computer role playing game (CRPG). This work focuses on generating layouts and plot structures to create novel game worlds through with a fitness function based on a transition graph that encodes pre-built game requirements. Tamari et al. (2019) focus on extracting action graphs for sequential decision making problems such as material science experiments and turn them into text-adventure games. Although these works use graph structures in order to constrain the generation of the game, we use these graph structures only to maintain coherence and focus on content creation.

Although there are works that attempt to automatically evaluate the creativity of the output of a generative process by computationally modeling potential human responses — such as with story telling (Purdy et al. 2018), etc. — we choose to rely on a human subject study based on the definition of creativity as presented in Boden (2007). Specifically we focus on the concepts of novelty and value, despite collecting data for other defined metrics as well. We use the definition of novelty stemming from the idea of p-creativity, i.e. a concept that is entirely new to a single agent — in this case a subject in our evaluation study. Value, as a component of computational creativity, however, is not defined concretely in Boden’s work for a general domain. Our definition of value in the context of text-adventure games relies on accomplishment or achievement.

Ammanabrolu et al. (2020) approach the problem of world generation in interactive fiction by turning linear stories into interactive worlds. They first extract a knowledge graph of the world from the story—containing locations, characters, and objects—and use that to generate the full game. Fan et al. (2019) leverage a crowdsourced dataset of fantasy text-adventure dialogues (Urbanek et al. 2019) to learn to generate interactive fiction worlds on the basis of of locations, characters, and objects. These works all focus on the problem of world generation in text-adventure games and do not contain objectives or quests—these systems are thus complimentary to ours.

Content Generation

In this section, we present Markov chains and neural language model based models to generate content, i.e. recipes, for our quests. Content generation for a quest in a text-adventure game, in this case a recipe, can be thought of as being equivalent to generating a sequence of events in which prior elements affect the probability of subsequent events. Markov chains present a simplified and well studied method to generate such content. Neural language models, designed to predict an element of a sequence conditioned on a given number of prior elements, let us generate sequences of events with more prior context—i.e. in the absence of the Markov assumption.
Markov Chains

Our first quest generation model is based on the use of Markov chains. This generation process consists of two steps. We first learn a weighted ingredient graph, a Markov chain, from a large scale knowledge base of recipes and then probabilistically walk along this graph to generate the instructions for the recipe.

Ingredient Graph Generating the recipe requires domain knowledge. For example, creating a recipe for peanut butter cookies requires an understanding that an ingredient like peanut butter fits well with eggs, flour, and sugar while something like fish does not. We represent this knowledge with an undirected graph of ingredients. Our ingredient graph is based off of recipes scraped from allrecipes.com. The raw, uncleaned dataset included over 20,000 recipes with over 4000 unique ingredients. A list of ingredients was extracted from each recipe, and each of these lists was converted into a set of ingredient pairs (Fig. 2). In total, there were 118,116 unique ingredient pairings, and 73,088 of those pairings (62%) only occurred once. We reduced the number of distinct ingredients from 4460 to 1703 by merging items with the same base ingredient and by replacing name-brand items with a generic equivalent.

Each of the nodes within the graph represents a possible ingredient, and weighted connections between these nodes represent how well the ingredients go together. The weight of each edge is the total number of occurrences of that ingredient pair within the recipe corpus. The edge connecting eggs and white sugar would have a weight of 3774 while the edge between hot milk and orange juice would have a weight of 1. Ingredient pairings that do not occur within the recipe corpus did not have an edge within this network, and this helped prevent our model from generating completely incoherent recipe pairings (e.g. hot sauce and baby food). Take the graph in Fig. 1 as an example. In this complete graph, all of the ingredients go well with each other except for fish and sugar, which is indicated by the low weight connection between them. The weak connection between sugar and fish suggest that they would likely not go well together in a recipe.

Instruction Generation With the ingredient graph created, we begin the process of instruction generation based on sub-graph mining and prior generative methods based on probabilistic graph walks (Fleishman 1978). We start by selecting an initial random ingredient ‘x’ weighted by its distribution in the graph.

\[ p(x_1) = \frac{\sum_{i=1}^{k} w(v_i, x_1)}{\sum_{i=1}^{k} \sum_{j=1}^{k} w(v_i, v_j)} \]  \hspace{1cm} (1)

We probabilistically select one of its neighbors based on the conditional frequency of the pair. Each iteration further computes conditional probabilities and selects them. We exclude all ingredients in which any bag of words token is contained by any other, ensuring that a variety of different ingredients are selected.

\[ \alpha = \begin{cases} 1 & \text{if } B_{x_i} \subseteq B_{x_{n+1}} \lor B_{x_{n+1}} \subseteq B_{x_i} \\ 0 & \text{else} \end{cases} \]  \hspace{1cm} (2)

In Eq. 2, \( B_{x_i} \) refers to the 1-gram bag of words model.

However, just computing complete conditional probabilities would remove the chance for entirely new combinations to emerge. Therefore, we calculate just the partial probability of having shared ingredients with a bias designed to favor such combinations.

\[ \beta = \left( \sum_{i=1}^{n} \text{Shared}(x_i, x_{n+1}) \right)^2 \]  \hspace{1cm} (3)

\[ \text{Shared}(x_1, x_2) = \begin{cases} 1 & w(x_1, x_2) > 0 \\ 0 & \text{else} \end{cases} \]  \hspace{1cm} (4)

This process repeated recursively to generate a recipe with the desired number of ingredients.

\[ p(x_{n+1}) = \frac{\sum_{i=1}^{n} \alpha \beta w(x_{n+1}, v_j)}{\sum_{j=1}^{k} w(x_i, v_j)} \]  \hspace{1cm} (5)

Finally, resultant combinations are referenced back against the original corpus to guarantee novelty in the result.

Neural Language Model

Our second technique uses a neural language model to generate both the ingredients for a recipe and the steps of the ingredients as well. We use the same knowledge base as
described in Sec. Ingredient Graph and train two separate language models: one to generate the ingredients, and the other to generate the recipe given a set of ingredients.

The first language model uses a simple 4-layer LSTM to generate a sequence of ingredients, treating all the words in a single ingredient as a single token. For example, “peanut butter” would be considered a single token in this model. We train this model using the sets of ingredients found in each recipe for the entire recipe dataset, with each set ending with an <EOI> or End of Ingredients tag. Once trained, the model then generates a sequence of ingredients until the <EOI> is reached using the top-k sampling technique (Holtzman et al. 2019).

To generate the actual recipe, we use GPT-2 (Radford et al. 2019) and fine-tune their pre-trained 345m parameter model on the recipe data. The data to fine-tune this model is designed to contain the recipe title, ingredients, and instructions in an unstructured text-form. Once this model has been fine-tuned, we use it to generate the title and instructions for the recipe conditioned on the ingredients generated by the first language model. The entire generated recipe consists of the ingredients, title, and instructions.

**Quest Assembly**

We now use the generated content—i.e. the recipe—to assemble a quest by grounding the generated ingredients and instructions in the game world. This requires us to first determine the structure of the game world and the locations of objects within this world in addition to transforming the set of generated instructions into executable actions. We use two types of semantically grounded knowledge graphs to represent this information: the object and action graphs.

The object graph is used to determine the structure of the world and the most likely locations of objects within this world in addition to transforming the set of generated instructions into executable actions. We use two types of semantically grounded knowledge graphs to represent this information: the object and action graphs.

The object graph is used to determine the structure of the world and the most likely locations of objects within this world. For example, we could have information that says that vegetables must be stored in a refrigerator. If a recipe requires carrots, then the carrots would automatically be placed in a refrigerator at the start of the game. This graph is constructed by hand and is built to make the game world and resulting quest as coherent as possible.

We construct object graphs for two different room layouts. The first, the one room (1R) map, consists of a kitchen as well as the objects and actions that exist within it. The second map, the five room (5R) map, is an extension of the first map and contains four additional rooms.

The object graph for the 1R map as shown in Fig. 3 is largely inspired by the simple, pre-built game provided within TextWorld (Côté et al. 2018). This object graph determines how and where objects are placed within the environment during game generation, and the action graph (Fig.6) dictates how generated instructions are transformed into executable actions in the game. The object graph was constructed logically: tools and utensils go in the drawer, meat and dairy belong in the refrigerator, and so on. Food item placements are deterministic and coherent. Vegetables always go in the refrigerator, and fruit always goes on the kitchen island. The action graph was also designed to prevent the player from conducting illogical actions.

The 5R map included a dining room, garage, backyard, and garden in addition to the kitchen (Fig. 4). The map (Fig. 5) is designed to maintain the same levels of coherency as the 1R map while allowing for more diverse gameplay, which could in turn lead to higher levels of perceived creativity. The additional rooms are selected based on their possible relationships to the domain of food and cooking, and each new room has its own unique objects that players can interact with. For example, the garage has an old refrigerator that can be used to store meat. These new rooms and objects also allow for dynamic food placement. Meat can be placed in one of two refrigerators, and fruits and vegetables can possibly be found in the garden. The existence of these new locations is not immediately clear to the player. The garage and backyard are additionally obscured...
by closed doors, adding to quest complexity. While the additional rooms and dynamic food placement allow for more diverse gameplay, they do not sacrifice coherency.

The action graph contains information regarding the affordances of the objects in the world and what kinds of objects are required to complete a generated instruction. For example, if a generated instruction tells us to prepare vegetables, i.e. cut them, then this graph tells us that there must be a knife somewhere in this world. This graph is partially extracted from static cooking guides online using a mixture of OpenIE (Angeli et al. 2015) and hand-authored rules to account for the irregularities of cooking guides. An example of an action graph is given in Fig. 6. A player can peel fruit and vegetables, for example, but cannot peel a steak. There are also strict rules on what tools are required for certain actions. A player can only cut something if they have a knife and can only peel something with a peeler. While this restricts how players can interact with the environment, it ultimately reinforces game coherency.

We also note that when generating the quests, both the Markov chain and the neural language model based generation systems use the object graph to determine object placement but only the Markov chain based model uses the action graph. This is because the instructions generated by the Markov chain model is in the form of a sequence of ingredients which then requires the action graph to determine the actions and additional objects required to turn this list of ingredients into a playable quest. The action graph would thus take an ingredient such as a carrot and determine first that it needs to be cut and that a knife is required for this task. The neural language model on the other hand already generates the full action, including potentially required objects, that can be executed and so does not make use of this graph.

**Experiments**

Our experiments were designed to compare perceived creativity and coherence, specifically testing our models in addition to factors such as complexity. We tested five types of designs: Human Designed (HD), Random Assignment (RA), Markov Chains Simple (MCS), Markov Chains Complex (MCC), and Language Model (LM). HD is simply what it sounds like: a game that was created by a person. In this game, a human—not associated with the research—creates both the ingredients and the instructions for a recipe and is additionally responsible for quest assembly, i.e. grounding the generated content in a given game world. We do not consider experience in designing text-adventure games when picking a human to create this game as this task can be performed even by novices given the easily understandable “home” theme of the game world. The game is manually crafted in terms of decided what ingredients to put where and what the final recipe would come together to be. RA is on the opposite end of the spectrum where, as the name suggests, everything is placed in a random location, and the recipe could be totally random with ingredients and instructions that might not normally be seen. MCS and MCC use our Markov chains approach to generate quest content. The difference between MCS and MCC are that the former has four ingredients involved in its recipe while the latter has eight. This was to vary the complexity to see how that affected perceived creativity. LM refers to the games generated using the recipes generated by the language model. We additionally had one-room and five-room variants for each of the models to test how the structure and length of the game would affect the players.

Automatically evaluating the creativity of the output of any computational generation process is a difficult task which requires concrete definitions of the metrics being used. We thus evaluate by deploying the game designs on Amazon Mechanical Turk for people to play and provide feedback. Specifically, they would play one randomly selected game from the 1 room layout and then fill out a survey for that game, and then play one randomly selected game from the 5 room layout and fill out an identical survey. Subjects were provided with a simple practice game that they could play beforehand to familiarize themselves with TextWorld and its interface. We had 75 total participants for the entire study and had an average of 15 people play each game. The only restrictions that we had for participants was that they had to be fluent in English—this was determined by means of pre-built restrictions on Amazon Mechanical Turk and game completion verification.

The users were asked questions pertaining to two metrics: coherence and creativity. We looked at creativity as a metric in the survey using the components of creativity as defined by Boden: novelty, surprise, and value. The survey detailed questions that measured our defined metrics, using Likert Scale values along a scale of 1-7. It posed questions such as “How coherent was the objective of the quest?”, “How original was the quest you played? 1: not at all novel, 7: exceptionally novel”, “Did you have a sense of accomplishment after completing the game? 1: no value, 7: extremely valuable”, “How unpredictable was the quest you played?” when measuring coherence, novelty, value, and surprise respectively. The other factors were also measured using similarly phrased questions. A one-way ANOVA test was then conducted followed by Tukey HSD post-hoc analysis to determine significance. The results of the raw scores for each group as well as the significant results between pairs of different models are presented below.
Results and Discussion

We present results for four metrics: coherence, unpredictability (or surprise), novelty (or originality), and value (or accomplishment) for each of the games. Additionally, we also show the $p$-value result of a one way ANOVA test for the distributions in each of the categories to determine statistical significance. This test tells us if the differences in the means across the different games are significant for each of the categories separately. The Tukey HSD post-hoc analysis further tells us which specific pairs of results are significant. We hypothesized that semantic grounding using the knowledge graph would enable our models to maintain coherence on par with the human designed games. Further, given the stochastic nature of our generative models, we further predicted that our models would also rate as being comparable in terms of creativity to the human designed games—with all models relatively outperforming the randomly generated games. We see below that these predictions hold.

We find that the results for each individual category are significant at $p < 0.05$ in all the cases. Additionally, all the specific pairwise comparisons we make are significant with $p < 0.1$. Table 1 presents some pairwise results along with the corresponding difference in scores and $p$-values. The rest of this section discusses these metrics in more detail.

Fig. 7 displays trends in the players' perception of coherence for each of the games. We first see that the one-room games were consistently rated to be more coherent than the five-room games, indicating that overall quest coherence—and thus the coherence of our generative system—degrades the longer and more complex the quest. Across the games, we see that the RA models were the considered to be the least coherent. The LM achieves a higher score than both of the Markov chain models and maintains coherence more easily than either. Most importantly, all of these methods are
### Table 1: Select pairwise results from the post hoc Tukey HSD test for each experiment.

<table>
<thead>
<tr>
<th>Group 1</th>
<th>Group 2</th>
<th>Mean diff</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coherence</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCC-5 RA-5</td>
<td>0.5333</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>LM-5 MCS-5</td>
<td>0.4392</td>
<td>0.066</td>
<td></td>
</tr>
<tr>
<td>LM-5 RA-5</td>
<td>1.0392</td>
<td>0.028</td>
<td></td>
</tr>
<tr>
<td>LM-1 RA-1</td>
<td>0.2843</td>
<td>0.042</td>
<td></td>
</tr>
<tr>
<td>Originality (novelty)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HD-1 LM-1</td>
<td>1.2292</td>
<td>0.057</td>
<td></td>
</tr>
<tr>
<td>HD-1 MCS-1</td>
<td>0.5347</td>
<td>0.10</td>
<td></td>
</tr>
<tr>
<td>HD-1 RA-1</td>
<td>1.5772</td>
<td>0.013</td>
<td></td>
</tr>
<tr>
<td>MCC-1 RA-1</td>
<td>1.0147</td>
<td>0.077</td>
<td></td>
</tr>
<tr>
<td>Unpredictability (surprise)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HD-1 MCC-1</td>
<td>0.9375</td>
<td>0.079</td>
<td></td>
</tr>
<tr>
<td>HD-1 RA-1</td>
<td>0.9228</td>
<td>0.071</td>
<td></td>
</tr>
<tr>
<td>LM-1 MCC-1</td>
<td>0.6667</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>LM-5 RA-5</td>
<td>-1.3255</td>
<td>0.026</td>
<td></td>
</tr>
<tr>
<td>MCC-1 MCS-1</td>
<td>-0.8611</td>
<td>0.086</td>
<td></td>
</tr>
<tr>
<td>MCC-5 RA-5</td>
<td>-1.2667</td>
<td>0.037</td>
<td></td>
</tr>
<tr>
<td>Accomplishment (value)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HD-1 LM-1</td>
<td>2.1042</td>
<td>0.022</td>
<td></td>
</tr>
<tr>
<td>HD-1 RA-1</td>
<td>1.6728</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td>HD-5 LM-5</td>
<td>0.8914</td>
<td>0.046</td>
<td></td>
</tr>
<tr>
<td>LM-1 MCC-1</td>
<td>-1.25</td>
<td>0.062</td>
<td></td>
</tr>
<tr>
<td>LM-1 MCS-1</td>
<td>-0.5</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>LM-5 MCS-5</td>
<td>-1.0196</td>
<td>0.06</td>
<td></td>
</tr>
<tr>
<td>MSc-5 RA-5</td>
<td>-1.2</td>
<td>0.05</td>
<td></td>
</tr>
</tbody>
</table>

comparable in coherence to the human-authored games, i.e. our semantically grounded knowledge graph ensures that coherence is not lost when generating content.

Originality (Fig. 8), which we use as a proxy to measure novelty, exhibits similar trends as surprise. The more complex, more coherent games are deemed more original. Despite being random, the RA games are seen to be less original than the rest of the games perhaps indicating that there is a link between perceptions of coherence and originality. The gaps in performance here are much less pronounced, however, and the Markov chain models slightly edge out the LM — with all three being comparable to the HD games.

Similarly, Fig. 9 describes how surprising the game was to the players. The difference between the one-room and five-room games here is much more pronounced. The players find the five-room, the longer and more complex game, much more surprising than their one-room counterparts, showing that complexity is an important factor in determining surprise. The LM achieves comparable performance to MCC and again they all perform as well as the HD games.

To measure value, or utility, in a text-adventure game, we asked the players if they felt a sense of accomplishment after finishing the game (Fig. 10). We see players reported a higher sense of accomplishment after finishing more complex games in general with the exception of the RA games, both of which performed poorly—likely due to them being relatively incoherent. We also note that the LM showed the highest values here, surpassing the HD games. We hypothesize that this might be due to the player having to perform a wider range of actions, some relatively unintuitive, that are not constrained by our action graph.

### Conclusions

We have demonstrated a framework to automatically generate cooking quests in a “home” themed text-adventure game, although our framework can be generalized to other themes as well. Quest generation in a given game world is a subset of the overall problem of generating entire text-adventure games. Content generated by both the Markov chains and the neural language models can be grounded into a given game world using domain knowledge encoded in the form of a knowledge graph. The models each excel on different metrics: the Markov chains model produces quests that are more surprising and novel while the neural language model offers greater value and coherence. We also note, however, that the neural language model requires less domain knowledge than the Markov chains and is thus potentially more generalizable to other themes and types of quests.

Our human subject study shows us that there is an inverse relationship between creativity and coherence but only when a certain threshold of coherence is passed. In other words, the less coherent a game the more creative it is, but incoherent games—such as those generated by the RA model—are perceived to be less creative. Furthermore, our automatically generated games consistently perform at least as well as human-designed games in this setting, both in terms of coherence and creativity—implying that the generative process can be automated without a loss in perceived game quality.

### References


Towards understanding situated natural language. In *Proceedings of the 2010 International Conference on Artificial Intelligence and Statistics*. 


Bisociative Literature-Based Discovery:
Lessons Learned and New Prospects

Nada Lavrač1,2, Matej Martinc1,4, Senja Pollak1, Bojan Cestnik3,1

1 Jožef Stefan Institute, Ljubljana, Slovenia
2 University of Nova Gorica, Nova Gorica, Slovenia
3 Temida d.o.o, Ljubljana, Slovenia
4 International Postgraduate School Jožef Stefan, Ljubljana, Slovenia

Abstract
The field of bisociative literature-based discovery aims at exploring scientific literature to reveal new discoveries based on yet uncovered relations between knowledge from different, relatively isolated fields of specialization. This paper outlines selected outlier-based literature mining approaches, which focus on finding outlier documents as means for finding unexpected links crossing different contexts. Selected approaches to bridging term detection through outlier document exploration are briefly outlined, together with the lessons learned from recent applications in medical and biological literature-based knowledge discovery. Finally, the paper addresses new prospects in bisociative literature-based discovery, emphasizing the use of advanced embeddings technology for cross-domain literature mining.

Introduction
Understanding complex phenomena and solving difficult problems often requires knowledge from different domains to be combined and cross-domain associations to be considered. While the concept of association is at the heart of several information technologies, including information retrieval and data mining e.g., association rule learning (Agrawal et al., 1996), scientific discovery usually requires to connect seemingly unrelated information from different domains. These kinds of bisociative context crossing associations, called bisociations (Koestler, 1964), are often needed for innovative discoveries.

In literature-based discovery (LBD) (Bruza and Weeber, 2008)—and in particular in cross-domain literature mining, which addresses knowledge discovery in two (several) initially separate document corpora—a crucial step is the identification of interesting bridging terms (b-terms) or links (b-links) that carry the potential of explicitly revealing the connections between the separate domains. Swanson (1990) and Smalheiser (1998) developed early LBD approaches to detecting interesting b-terms to uncover the possible cross-domain relations among previously unrelated concepts. For example, the ARROWSMITH online system, developed by Smalheiser and Swanson (1998), takes as input two sets of titles of scientific papers from disjoint domains (disjoint document corpora) A and C, and lists terms that are common to A and C; the resulting bridging terms (b-terms) are further investigated by the user for their potential to generate new scientific hypotheses.1 Their approach, known as the ‘ABC model of knowledge discovery’, is based on the closed discovery setting (Weeber et al., 2001), where two initially separate domains A and C are specified by the user at the beginning of the discovery process, and the goal is to search for bridging concepts (terms) b in B to validate the hypothesized connection between A and C. The closed discovery setting addressed in this paper is illustrated in Figure 1.

Figure 1: Closed discovery process defined by Weeber et al. (2001).

Inspired by early Swanson’s and Smalheiser’s work, literature mining approaches were further developed and successfully applied to different problems, such as finding associations between genes and diseases (Hristovski et al., 2005), diseases and chemicals (Yetisgen-Yildiz and Pratt, 2006), and many others. Holzinger et al. (2013) describe several quality-oriented web-based tools for the analysis of biomedical literature, which include the analysis of terms (biomedical entities such as disease, drugs, genes, proteins and organs) and provide concepts associated with a given term. The work of Kastrin, Rindflesch, and Kauschke (2013) further develops the concept of associated terms, and the use of cross-domain knowledge.

In the ABC model, uppercase letter symbols A, B and C are used to represent concepts (or sets of terms), and lowercase symbols a, b and c to represent single terms.
tovski (2014) is complementary to other LBD approaches, as it uses different similarity measures (such as common neighbors, Jaccard index, and preferential attachment) for link prediction of implicit relationships in the Semantic MEDLINE network. A comprehensive survey of modern literature-based discovery approaches in biomedical domain can be found in Sebastian, Siew, and Orimaye (2017) and Gopalakrishnan et al. (2019).

This work follows the line of research in two closely related areas, which provide computational tools that act as creative assistants to support human creativity: literature-based discovery, described in some detail above, as well as bisociative knowledge discovery, where—according to Berthold (2012)—two concepts are bisociated if there is no direct, obvious evidence linking them and if one has to cross different domains to find the link, where a new link must provide some novel insight into the problem addressed. In the context of this paper, both research areas address the same computational creativity task of bridging term (b-term) detection when exploring the connections between two different domains of interest.

More generally, bisociative knowledge discovery addresses a data mining task where two or more domains of interest are searched for bisociative links or individual bridging concepts (i.e. individual context bridging terms). Bisociative knowledge discovery differs from more standard discovery science and associative data mining approaches, like the standard association rule learning (Agrawal et al., 1996), which focus on knowledge discovery within a given domain of interest. Notably, the ability of literature-based discovery and bisociative knowledge discovery methods and software tools that aim to support the experts in their knowledge discovery processes—especially in searching for yet unexplored connections between different domains—is becoming increasingly important.

This paper outlines selected approaches to cross-domain literature mining that support the expert in searching for hidden links connecting two seemingly unrelated domains. The next section below outlines our early approaches to cross-domain literature mining via outlier document detection and exploration (Petrič et al., 2012; Sluban et al., 2012), together with the lessons learned from their past applications in medical literature mining. The subsequent section presents a more recent implementation of the outlier-based approach to LBD (Cestnik et al., 2017), which implements ensemble-based term ranking using an ensemble of six elementary heuristics for b-term evaluation, and incorporates also the human-computer interface (HCI) of the CrossBee b-term detection system (Juršič et al., 2012), together with the lessons learned from the recent LBD applications. The literature based discovery workflow implemented in TextFlows (Perovšek et al., 2016), acting as the enabling technology for implementing the developed cross-domain link discovery approach, is also briefly mentioned. The last sections propose some future research directions based on the lessons learned from the current text mining research, including the idea of a novel LBD framework motivated by the recent word embedding technology. The paper concludes with a summary and some plans for further work.

Outlier-based LBD: Early results and lessons learned

Outliers, characterized by their properties of being infrequent or unusual, may represent unexpected events, entities, items or documents. Early research in LBD has focused on the identification and exploration of outlier documents since they frequently embody new information that is often hard to explain in the context of existing mainstream knowledge. The LBD research (Petrič et al., 2012) and (Sluban et al., 2012) suggests that bridging terms are more frequent in documents that are in some sense different from the majority of documents in a given domain.

The outlier-based approach to LBD proposed by Petrič et al. (2012) uses document clustering to find outlier documents. The approach consists of two steps. In the first step, the OntoGen clustering algorithm (Fortuna, Grobelnik, and Mladenović, 2006) is applied to cluster the merged document set A ∪ C, consisting of documents from two domains A and C. The result of unsupervised clustering are two document clusters: A′ = Classified as A (i.e. documents from A ∪ C classified as A), and C′ = Classified as C (i.e. documents from A ∪ C classified as C). In the second step of outlier detection, clusters A′ and C′ are further separated, each into two clusters, based on the documents’ original labels A and C. As a result, a two-level tree hierarchy of clusters is generated, illustrated in Figure 2.

Lesson Learned 1: Potential of outlier documents. The hypothesis that outlier documents have the potential to improve the effectiveness of bridging term detection was tested on the migraine-magnesium (Swanson, Smallheiser, and Torvik, 2006) and autism-calcineurin (Petrič et al., 2009) domain pair datasets, which have lists of concept bridging terms (b-terms) confirmed by the medical experts. The experimental results obtained by using OntoGen confirm the hypothesis that most bridging terms appear in outlier documents and that by considering only outlier documents the search space for b-term identification can be largely reduced.

This lesson—that outlier documents have the potential for improving the effectiveness of bridging term detection—was reconfirmed in the work of Sluban et al. (2012), exploring a classification filtering approach to outlier detection, which was tested on the same domain pair data sets, migraine-magnesium (Swanson, Smalheiser, and Torvik, 2006) and autism-calcineurin (Petrič et al., 2009) domain, which have lists of bridging terms (b-terms) confirmed by the medical experts. Sluban et al. (2012) proposed to detect outlier documents using classification algorithms for classification noise filtering, first suggested by Brodley and Friedl (1999). Having documents from two domains of interest A and C, Sluban et al. (2012) first trained an ensemble classifier that distinguishes between the documents of these domains, and use the ensemble classifier to classify all the documents. The miss-classified documents were declared as outliers, since—according to the classification model—they do not belong to their domain (class label) of origin. These outliers can be interpreted as borderline documents as they were considered by the model to be more similar to the other domain.
Figure 2: Target domain documents from disjoint literatures A and C, clustered according to the proposed OntoGen’s two step approach, first using unsupervised and then supervised clustering to obtain outlier documents O(A) and O(C) of literatures A and C, respectively. The figure illustrates the outlier document detection approach as implemented in OntoGen, addressing the outlier detection framework, conceptually explained in Figure 3.

Figure 3: Detecting outliers of a domain pair dataset A ∪ C, using a document classification approach by Sluban et al. (2012).

than to their original domain, and can be regarded as bridging documents between the two domains. In other words, if an instance of class A is classified in the opposite class C, it is considered an outlier of domain A, and vice versa. The two sets of outlier documents were denoted with O(A) and O(C), illustrated in Figure 3.

The experimental results obtained by Sluban et al. (2012) showed that the sets of detected outlier documents are relatively small—including less than 5% of the entire datasets—and that they contain a great majority of bridging terms previously identified by medical experts, which was significantly higher than in same-sized random document subsets. These results are summarized in Figure 4.

These experimental results indicate that it is justified that the search for b-terms can be focused on outlier documents, which contain a large majority of b-terms. Consequently, by focusing the exploration on outlier documents, the effort needed for finding cross-domain links is substantially reduced, as it requires to explore a much smaller subset of documents, where a great majority of b-terms are present and more frequent.

Figure 4: Presence of b-terms in the detected outlier sets of two domain pair datasets.
Outlier-based LBD: Current applications and lessons learned

When applying OntoGen on the documents of the new application domain using the Alzheimer's disease-gut microbiome domain pair (Cestnik et al., 2017), the OntoGen method uses domains A and C, and builds a joint document set $A \cup C$. With this intention, two individual sets of documents (e.g., titles, abstracts or full texts of scientific articles), one for each domain under research (namely, literature A on Alzheimer’s disease and literature C on gut microbiome), were automatically retrieved from the PubMed database. A cluster hierarchy was constructed from the dataset of 17,863 papers with OntoGen. Two first-level clusters are labeled with the OntoGen suggested keywords ad, abeta, cognitive and microbiota, gut, intestine. Four second-level sub-clusters separate documents according to their original search keywords for Alzheimers disease and gut microbiome, as illustrated in Figure 5.

Lesson Learned 2: Excluding intersecting documents.

In the Alzheimer’s disease-gut microbiome LBD application, the initial document set $A \cup C$ consisted of some documents, which were in the intersection of A and C, meaning that a few documents were retrieved from PubMed by both of the two separate queries for domain A (i.e. Alzheimer) and C (i.e. gut OR intestinal) AND (microbiota OR bacteria), which was surprising. After carefully inspecting these documents (as these documents could contain the b-terms representing a solution to the problem, which proved not to be the case) it was realized that keeping them in the $A \cup C$ document set was problematic. As a result, the documents that were retrieved by both queries were eliminated, resulting in 17,863 documents kept in the $A \cup C$ document set used for further exploration.

Lesson Learned 3: Selecting only outlier documents.

The hypothesis that the search for bridging terms can be reduced to manageable subsets of documents was confirmed in our experiments. In the Alzheimer’s disease-gut microbiome LBD application using OntoGen for outlier document detection, the space of documents used for b-term exploration was further reduced from the set of 17,863 documents to two subsets of outlier documents, i.e. to only 154 gut microbiome papers and 428 Alzheimer’s disease related papers, considered as outliers in their own domain, leading to the selection of only 582 documents for further inspection.

Lesson Learned 4: Expert revision of b-terms list. The hypothesis that b-terms selected from outlier documents can be further reduced with expert knowledge was confirmed in our experiments. By processing the remaining 582 outlier documents, we used CrossBee (Juršič et al., 2012) to extract 4,723 terms as potential b-terms connecting the two domains. In b-term exploration all the terms were considered and not just the medical ones, except that a list of 523 English stop words was used to filter out meaningless words, and English Porter stemming was applied, which helped us to focus on medically interesting b-terms. Even though the list of potential bridging terms was ordered according to the ensemble-heuristics estimated bridging terms potential, browsing and analyzing the terms from the list still presented a substantial burden for the domain expert. To further reduce the size of the potential b-term list, the collaborating domain expert prepared a list of 289 domain terms of her own research interest. This list included common terms and specific molecular factors and pathways, which were manually identified in titles, abstracts, and keywords from 42 papers obtained from PubMed search query (gut AND Alzheimer), 55 of which appeared also among the 4,723 terms extracted by CrossBee. During the evaluation phase, the relevant papers for each b-term candidate were reviewed and searched for potential clues justifying further investigation, resulting from relevant b-term discoveries confirmed by the domain expert (Cestnik et al., 2017).

Compared to outlier document detection using OntoGen, an upgraded methodology proposed by Cestnik et al. (2017) was implemented in a reusable outlier based LBD methodology in a web-based text mining platform TextFlows.

---

2Their inclusion in the document set would have violated the assumption of literature-based discovery and bisociative knowledge discovery frameworks, which assume that the explored literature domains A and C are disjoint; if this assumption were violated, the methodology would fail due to biased heuristics calculations.

3Elsa Fabretti

4http://textflows.org
(Perovšek et al., 2016) that allowed us to construct and execute advanced text mining workflows. The workflow shown in Figure 6 consists of seven steps implemented as subprocesses. The connections between subprocesses represent the flow of documents from one subprocess to another. In overview, steps 1-3 represent the outlier detection part, and steps 4-7 represent cross-domain exploration for b-term detection.

Lesson Learned 5: TextFlows workflow helping experts.

In the experiments using the TextFlows workflow, the NoiseRank ensemble-based outlier detection approach (Sluban, Gamberger, and Lavrač, 2013) as implemented in TextFlows was used. The goal of the first three steps (using first three workflow widgets) of the methodology is to effectively extract a set of outlier documents from the whole corpus of input documents. Consequently, by decreasing the size of the input set of documents the second phase becomes more focused, efficient and effective. In the last four steps of the workflow in Figure 6 components that constitute the CrossBee HCI interface (Juršič et al., 2012) are executed to conduct expert-guided b-term analysis. Here, the goal is to further prepare the input documents for b-term visualization and exploration. Note that in this step the role of the domain expert is crucial.

Lesson Learned 6: Term filtering and synonyms matter.

In recent LBD experiments, using plant defence-circadian rhythm domain pair, the goal was to identify potentially interesting new daily regulated mechanisms that are responsible for plant defence. After obtaining 5,412 documents from PubMed containing complete articles (2,483 from plant defence and 2,929 from circadian rhythm), 0.5% documents shorter than 20 characters (mostly empty contents) and longer than 97,500 characters (containing many different articles in proceedings) were removed. Then, 12 duplicates that were present in both domains (as in Lesson Learned 2) were eliminated. The crucial, although simple and straightforward, step in this experiment was the replacement of gene names with synonyms gathered in previous research projects (22,265 gene names mapped into 7,863 synonyms). In addition, the documents were optionally pre-processed to keep only gene-related terms (included in synonym list and from the gene dictionary containing additional 6,083 gene names), which resulted in a substantial reduction of the input file size (from 200 MB to 28 MB).

Current research lessons

Future work, aimed at improving the effectiveness of bridging term detection in cross-domain literature mining, will be performed in several directions. It will be based on the lessons learned in the current research: using embeddings for representation learning used in document clustering, using ontologies for term enrichment in cross-domain document exploration, and using network analysis for cross-domain heterogeneous information network exploration.

Related Lesson 1. The use of background knowledge remains largely unexploited in text classification and clustering. Word taxonomies can easily be exploited as means for constructing new semantic features, which can be used in text representation learning to improve the performance and robustness of the learned models. Consequently, recently developed tax2vec algorithm could be used for constructing taxonomy based features to improve the results of document clustering and classification.

Related Lesson 2. Given that documents can be easily transformed into graphs (e.g., graphs constructed from subject-verb-object triplets), network analysis approaches can prove to be fruitful for bridging term detection (e.g., community detection and finding bridging nodes in graphs between subgraphs representing the detected communities). In addition to network analysis approaches, novel graph embedding approaches could also be used in this context.
Related Lesson 3. Instead of using the standard TF-IDF (Term Frequency Inverse Document Frequency) weighted Bag Of Words vector representations of text documents, which was used in the past LBD research outlined in this paper (Petríč et al., 2012; Sluban et al., 2012; Jursič et al., 2012; Cestnik et al., 2017), the current research in EMBEDDIA\(^5\) indicates that representation learning using embeddings is much more effective than using the standard TF-IDF Bag of Words document representation. Consequently, improved clustering results can be expected using contemporary embedding approaches such as word2vec, doc2vec or Bert.

Complying with Related Lesson 3, this section proposes a novel approach to bisociative discovery between two separate domains A and C, using the power of word embeddings. Word embeddings are vector representations of words: each word is assigned a vector of several hundred dimensions. These are usually obtained via training algorithms such as word2vec (Mikolov et al., 2013), GloVe (Pennington, Socher, and Manning, 2014) or FastText (Bojanowski et al., 2017), which characterize the word based on the lexical context in which it appears. These representations improve performance in a wide range of automated text processing tasks, partly because they capture a degree of semantics. They can also capture regularities beyond simple relatedness, such as analogies (Mikolov, Yih, and Zweig, 2013); for example, the vector-space relation between Madrid and Spain is very similar to that between Paris and France.

In a closed literature based discovery setting, we are interested if a specific relation between two concepts (a1 and a2) in the first domain A could also be found between concepts x and c in the second domain C, where concept c is given in advance and x is the new concept that we are trying to find. More formally, this can be written in a form of an analogy (i.e. bisociation) between two separate domains A and C:

\[
al_1 \text{ rel } a_2 == x \text{ rel } c
\]

In the embedding space, this analogy translates to the following equation between embeddings:

\[
x = \text{embedding } a_1 + \text{embedding } a_2 - \text{embedding } c
\]

Finally, once x is calculated, we need to find a set of concepts from the second domain that have an embedding representation most similar to x according to the cosine similarity.

In the context of computational creativity research based on bisociation (Koestler, 1964), bisociative patterns that are searched and explored include: bridging concepts, bridging graphs, and bridging by structural similarity (Kötter and Berthold, 2012). The embeddings-based bisociative knowledge discovery approach described above addresses the latter, most complex setting of bridging by structural similarity, defined as follows:

**Bridging by structural similarity.** This is the most complex kind of bisociation, whereby in a bisociative network representation of concepts, subsets of concepts in each domain share structural similarities. Bisociations based on structural similarity are represented by relations and/or sub-graphs of two different, structurally-similar domains Kötter and Berthold (2012).

This type of bisociation is according to Kötter and Berthold (2012) the most abstract pattern with the potential for new cross-domain discoveries, which e.g., graph similarity methods can identify.

In our preliminary experiments using plant defence-circadian rhythm domain pair, where the goal was to identify potentially interesting new daily regulated mechanisms that are responsible for plant defence, we employed FastText embeddings (Bojanowski et al., 2017), in which a word is represented as an average of its character n-grams. This allows the model to leverage both semantic and morphological information, which is useful in a setting with small domain corpora containing less semantic information, since morphological similarity in many cases translates to semantic relatedness. Separate embedding models were trained for domains A and C and then aligned into a common vector space by using a supervised approach that relies on a training dictionary of identical words from both domains, used as anchor points to learn a mapping from the source to the target space with Procrustes alignment (Conneau et al., 2017).

**Conclusions**

This paper addresses the field of scientific computational creativity, in particular bisociative literature-based discovery. The paper mostly focusing on finding outlier documents as means for finding unexpected links crossing different contexts. Selected approaches to bridging term detection through outlier document exploration are briefly outlined, together with the lessons learned from recent applications in medical and biological literature-based knowledge discovery. Finally, the paper addresses new prospects in bisociative literature-based discovery, emphasizing the use of advanced embeddings technology for cross-domain literature mining.

In future work we will further explore embeddings-based LBD both in the closed and in the open LBD settings. We will also introduce additional user interface options for data visualization and exploration as well as advance the term ranking methodology by adding new sophisticated heuristics, which will take into account also the semantic aspects of the data. Finally, by using the recent word embedding technology, we aim to implement a novel bisociative knowledge discovery setting of bridging by structural similarity.

**Acknowledgements**

This work was supported by Slovenian Research Agency grant P-0103 and the European Unions Horizon 2020 research and innovation programme under grant agreement No 825153, project EMBEDDIA (Cross-Lingual Embeddings for Less-Represented Languages in European News Media).

**References**

Paranoid Transformer:
Reading Narrative of Madness as Computational Approach to Creativity

Yana Agafonova
Higher School of Economics,
Saint-Petersburg

Alexey Tikhonov
Yandex,
Berlin

Ivan P. Yamshchikov
Max Planck Institute
for Mathematics in the Sciences,
Leipzig
ivan@yamshchikov.info

Abstract
This paper revisits the receptive theory in context of computational creativity. It presents a case study of a Paranoid Transformer — a fully autonomous text generation engine with raw output that could be read as the narrative of a mad digital persona without any additional human post-filtering. We describe technical details of the generative system, provide examples of output and discuss the impact of receptive theory, chance discovery and simulation of fringe mental state on the understanding of computational creativity.

Introduction
The studies of computational creativity in the field of text generation commonly aim to represent a machine as a creative writer. Although text generation is broadly associated with a creative process, it is based on linguistic rationality and the common sense of the general semantics. In (Yamshchikov et al. 2019), authors demonstrate that if a generative system learns a better representation for such semantics, it tends to perform better in terms of human judgment. However, since averaged opinion could hardly be a beacon for human creativity, is its’ usage feasible regarding computational creativity?

The psychological perspective on human creativity tends to apply statistics and generalizing metrics to understand its object (Rozin 2001; Yarkoni 2019), so creativity becomes introduced through particular measures, which is epistemologically suicidal for aesthetics. While both creativity and aesthetics depend on judgemental evaluation and individual taste that depends on many aspects (Hickman 2010; Melchionne 2010), the concept of perception has to be taken into account, when talking about computational creativity.

The variable that is often underestimated in the mere act of meaning creation is the reader herself. Although the computational principles are crucial for text generation, the importance of a reading approach to generated narratives is to be revised. What is the role of the reader in the generative computational narrative? This paper tries to address these two fundamental questions presenting an exemplary case study.

The epistemological disproportion between common sense and irrationality of the creative process became the fundamental basis of the research. It encouraged our interest in reading a generated text as a narrative of madness. Why do we treat machine texts as if they are primitive maxims or well known common knowledge? What if we read them as narratives with the broadest potentiality of meaning like insane notes of asylum patients? Would this approach change the text generation process?

In this paper, we present the Paranoid Transformer, a fully autonomous text generator that is based on a paranoiac-critical system and aims to change the approach to reading generated texts. The critical statement of the project is that the absurd mode of reading and the evaluation of generated texts enhances and changes what we understand under computational creativity. Another critical aspect of the project is that Paranoid Transformer resulting text stream is fully unsupervised. This is a fundamental difference between the Paranoid Transformer and the vast majority of text generation systems presented in the literature that are relying on human post-moderation, i.e., cherry-picking.

Originally, Paranoid Transformer was represented on the National Novel Generation Month contest1 (NaNoGenMo 2019) as an unsupervised text generator that can create narratives in a specific dark style. The project has resulted in a digital mad writer with a highly contextualized personality, which is of crucial importance for the creative process (Veale 2019).

Related Work
There is a variety of works related to the generation of creative texts like the generation of poems, catchy headlines, conversations, and texts in particular literary genres. Here we would like to discuss a certain gap in the field of creative text generation studies and draw attention to the specific reading approach that can lead to more intriguing results in terms of computational creativity.

The interest in text generation mechanisms is rapidly growing since the arrival of deep learning. The there are various angles from which researcher approach text generation. For example, (van Stegeren and Theune 2019) and (Alnajjar, Leppänen, and Toivonen 2019) study generative models that could produce relevant headlines for the news publications. A variety of works study stylization potential of generative models either for prose, see (Jhamtani et al. 2019).

---

1https://github.com/NaNoGenMo/2019
Generative poetry dates back as far as (Wheatley 1965) along with other early generative mechanisms and has various subfields at the moment. Generation of poems could be addressed following specific literary tradition, see (He, Zhou, and Jiang 2012; Yan et al. 2016; Yi, Li, and Sun 2017); could be focused on the generation of topical poetry (Ghazvininejad et al. 2016); could be centered around stylization that targets a certain author (Yamshchikov and Tikhonov 2019) or a genre (Potash, Romanov, and Rumshisky 2015). For a taxonomy of generative poetry techniques, we address the reader to (Lamb, Brown, and Clarke 2017).

The symbolic notation of music could be regarded as a subfield of text generation, and the research of computational potential in this context has an exceptionally long history. To some extent, it holds a designated place in the computational creativity hall of fame. Indeed, at the very start of computer-science Ada Lovelace already entertains a thought that an analytical engine can produce music on its own. (Menabrea and Lovelace 1842) state: "Supposing, for instance, that the fundamental relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptations, the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent." For an extensive overview of music generation mechanisms, we address the reader to (Briot, Hadjeres, and Pachet 2019).

One has to mention a separate domain related to different aspects of the ‘persona’ generation. These could include relatively well-posed problems such as the generation of biographies out of the structured data, see (Lebret, Grangier, and Auli 2016), or open-end tasks for the personalization of dialogue agent, dating back to (Weizenbaum 1966). With the rising popularity of chat-bots and the arrival of deep learning, the area of persona-based conversation models (Li et al. 2016) is growing by leaps and bounds. The democratization of generative conversational methods provided by open-source libraries such as (Burtsyn et al. 2018; Shiv et al. 2019) fuels further advancements in this field.

However, the majority of text generation approaches are chasing the generation as the significant value of such algorithms, which makes the very concept of computational creativity seem less critical. Another major challenge is the presentation of the algorithms’ output. Vast majority of results on natural language generation either do not imply that generated text has any artistic value, or expect certain post-processing of the text to be done by a human supervisor before the text is presented to the actual reader. We believe that the value of computational creativity is to be restored by shifting the researcher’s attention from generation to the process of framing the algorithm (Charney, Pease, and Colton 2012). We show that such shift it possible since the generated output of Paranoid Transformer does not need any additional laborious manual post-processing.

The most reliable framing approaches are dealing with attempts to clarify the algorithm by providing the context, describing the process of generative acts, and making calculations about the generative decisions (Cook et al. 2019). In this paper, we suggest that such an unusual framing approach as the obfuscation of the produced output could be quite profitable in terms of increasing the number of interpretations and enriching the creative potentiality of generated text.

Obfuscated interpretation of the algorithm’s output methodologically intersects with the literary theory that deals with the reader as the key figure responsible for the meaning. In this context, we aim to overcome disciplinary borderline and create bisociative knowledge, which develops the fundamentals of computational creativity (Veale and Cardoso 2019). This also goes in line with the ideas of (Ohsawa 2003; Abe 2011) regarding obfuscation as a mode of reading generated texts that the reader either commits voluntarily or is externally motivated to switch gears and perceive generated text in such mode. This commitment implies a chance discovery of potentially rich associations and extensions of possible meaning.

How exactly can literary theory contribute to computational creativity in terms of the text generation mechanisms? As far as the text generation process implies an incremental interaction between neural networks and a human, it inevitably presupposes critical reading of the generated text. This reading brings a lot in the final result and comprehension of artificial writing. In Literature studies, the process of meaning creation is broadly discussed by hermeneutical philosophers, who treated the meaning as a developing relationship between the message and the recipient, whose horizons of expectations are constantly changing and enriching the message with new implications (Gadamer 1994; Hirsch 1967).

The importance of reception and its difference from author’s intentions was convincingly demonstrated and articulated by the so-called Reader-response theory, a particular branch of the Receptive theory that deals with verbalized receptions. As Stanley Fish, one of the principal authors of the approach, puts it, the meaning does not reside in the text but in the mind of the reader (Fish 1980). Thus, any text may be interpreted differently, depending on the reader’s background, which means that even an absurd text could be perceived as meaningful under specific circumstances. The same concept was described by (Eco 1972) as so-called aberrant reading and implied that the difference between intention and interpretation is a fundamental principle of cultural communication. It is often the shift in interpretative paradigm that makes remarkable works of art to be dismissed by most at first like Picasso’s Les Demoiselles d’Avignon that was not recognized by artistic society and was not exhibited for nine years since it had been created.

One of the most recognizable literary abstractions in terms of creative potentiality is the so-called ‘romantic mad poet’ whose reputation was historically built on the idea that genius would never be understood (Whitehead 2017). Madness in terms of cultural interpretation is far from its psychiatric meaning and has more in common with the historical concept of a marginalized genius. Mad narrator was chosen as a literary employ for the Paranoid Transformer to extend the interpretative potentiality of the original text that could...
be not ideal in formal terms, on the other hand, it could be attributed to an individual with exceptional understanding of the world, which gives more linguistic freedom to this individual for expressing herself and more freedom in interpreting her messages. The anthropomorphization of the algorithm makes the narrative more personal, which is as important as the personality of a recipient in the process of meaning creation (Dennett 2014). The self-expression of the Paranoid Transformer is enhanced by introducing a nervous handwriting that amplifies the effect and gives more context for interpretation. In this paper, we show that treating the text generator as a romantic mad poet gives more literary freedom to the algorithm and generally improves the text generation. The philosophical basis of our approach is derived from the idea of creativity as an act of transgressing the borderline between conceptual realms. Thus, the dramatic conflict between computed and creative text could be solved by extending the interpretative horizons.

Model and Experiments
The general idea behind the Paranoid Transformer project is to build a ‘paranoid’ system based on two neural networks. The first network (Paranoid Writer) is a GPT-based (Radford et al. 2019) tuned conditional language model, and the second one (Critic subsystem) uses a BERT-based classifier (Devlin et al. 2019) that works as a filtering subsystem. The critic selects the ‘best’ texts from the generated stream of texts that Paranoid Writer produces and filters the ones that it deems to be useless. Finally, an existing handwriting synthesis neural network implementation is applied to generate a nervous handwritten diary where a degree of shakiness depends on the sentiment strength of a given sentence. This final touch further immerses the reader into the critical process and enhances the personal interaction of the reader with the final text. Shaky handwriting frames the reader and, by design, sends her on the quest for meaning.

Generator Subsystem
The first network, Paranoid Writer, uses an OpenAI GPT (Radford et al. 2019) architecture implementation by huggingface. We used a publicly available model that was already pre-trained on a huge fiction BooksCorpus dataset with approximately 10K books with 1B words.

The pre-trained model was fine-tuned on several additional handcrafted text corpora, which altogether comprised approximately 50Mb of text for fine-tuning. These texts included:

- a collection of Crypto Texts (Crypto Anarchist Manifesto, Cyphernomicon, etc.);
- a collection of fiction books from such cyberpunk authors as Dick, Gibson, and others;
- non-cyberpunk authors with particular affinity to fringe mental prose, for example, Kafka and Rumi;
- transcripts and subtitles from some cyberpunk movies and series such as Bladerunner;
- several thousands of quotes and fortune cookie messages collected from different sources.

During the fine-tuning phase, we have used special labels for conditional training of the model:

- QUOTE for any short quote or fortune, LONG for others;
- CYBER for cyber-themed texts and OTHER for others.

Each text got two labels, for example, it was LONG+CYBER for Cyphernomicon, LONG+OTHER for Kafka, and QUOTE+OTHER for fortune cookie messages. Note, there were almost no texts labeled as QUOTE+CYBER, just a few nerd jokes. The idea of such conditioning and the choice of texts for fine-tuning was rooted in the principle of reading a madness narrative discussed above. The obfuscation principle manifests itself in the fine-tuning on the short aphoristic quotes and ambivalent fortune cookies. It aims to enhance the motivation of the reader and to give her additional interpretative freedom. Instrumentally the choice of the texts was based on two fundamental motivations: we wanted to simulate a particular fringe mental state, and we also were specifically aiming into short diary-like texts to be generated in the end. It is well known that modern state-of-the-art generative models are not able to support longer narratives yet can generate several consecutive sentences that are connected with one general topic. QUOTE/LONG label allowed us to control the model and to target shorter texts during the generation. Such short ambivalent texts could subjectively be more intense. At the same time, inclusion of longer texts in the fine-tuning phase allowed us to shift the vocabulary of the modal even further toward a desirable ‘paranoid’ state. We also were aiming into some proxy of ‘self-reflection’ that would be addressed as a topic in the resulting ‘diary’ of the paranoid transformer. To push the model in this direction, we introduced cyber-themed texts. As a result of these two choices, in generation mode, the model was to generate only QUOTE+CYBER texts. The raw results were already promising enough:

let painting melt away every other shred of reason and pain, just lew
the paint to move thoughts away from blizzes in death. let it dry out,
and turn to cosmic delights, to laugh
on the big charms and saxophones and
fudatron steames of the sales titanium.
we are god’s friends, the golden hands
on the shoulders of our fears. do
you knock my cleaning table over? i
snap awake at some dawn. the patrons
researching the blues instructor’s
theories around me, then give me a
glass of jim beam. boom!

However, this was not close enough to any sort of creative process. Our paranoid writer had graphomania too. To amend this mishap and improve the resulting quality of the texts, we wanted to incorporate additional automated filtering.
Heuristic Filters
As a part of the final system, we have implemented heuristic filtering procedures alongside with a critic subsystem.

The heuristic filters were as follows:

• reject the creation of new, non-existing words;
• reject phrases with two unconnected verbs in a row;
• reject phrases with several duplicating words;
• reject phrases with no punctuation or with too many punctuation marks.

The application of this script cut the initial text flow into a subsequence of valid chunks filtering the pieces that could not pass the filter. Here are several examples of such chunks after heuristic filtering:

a slave has no more say in his language but he has to speak out!
the doll has a variety of languages, so its feelings have to fill up some time of the day - to - day journals.
the doll is used only when he remains private. and it is always effective.
leave him with his monk - like body.
a little of technique on can be helpful.

To further filter the stream of such texts, we implemented a critic subsystem.

Critic subsystem
We have manually labeled 1 000 of generated chunks with binary labels GOOD/BAD. We marked a chunk as BAD in case it was grammatically incorrect or just too dull or too stupid. The labeling was profoundly subjective. We marked more disturbing and aphoristic chunks as GOOD, pushing the model even further into the desirable fringe state of paranoia simulation. Using these binary labels, we have fine-tuned a pre-trained publicly available BERT classifier3 to predict the label of any given chunk.

Finally, a pipeline that included the Generator subsystem, the heuristic filters, and the Critic subsystem produced the final results:

a sudden feeling of austin lemons, a gentle stab of disgust. i’m what i’m humans whirl in night and distance.
we shall never suffer this. if the human race came along tomorrow, none of us would be as wise as they already would have been. there is a beginning and an end.
both of our grandparents and brothers are overdue. he either can not agree or he can look for someone to blame for his death.
he has reappeared from the world of revenge, revenge, separation, hatred.

The resulting generated texts were already thought-provoking and allowed reading a narrative of madness, but we wanted to enhance this experience and make it more immersive for the reader.

Nervous Handwriting
In order to enhance the personal aspect of the artificial paranoid author, we have implemented an additional generative element. Using implementation4 for handwriting synthesis from (Graves 2013), we have generated handwritten versions of the generated texts. Bias parameter was used to make the handwriting shakier if the generated text’s sentiment was stringer. Figures 1–3 show several final examples of the Paranoid Transformer diary entries.

Figure 1 demonstrates that the length of the entries can differ from several consecutive sentences that convey a longer line of reasoning to a short, abrupt four-words note.

Figure 2 illustrates typical entry of ‘self-reflection’. The text explores the narrative of dream and could be paralleled with a description of an out-of-body experience (Blanke et al. 2004) generated by the predominantly out-of-body entity.

---

3https://github.com/huggingface/transformers#model-architectures
4https://github.com/sjvasquez/handwriting-synthesis
Figure 3: Some examples of Paranoid Transformer diary entries. Typical entries with destructive and ostracised motives.

Figure 3 illustrates typical entries with destructive and ostracised motives. This is an exciting side-result of the model that we did not expect. The motive of loneliness is recurring in the Paranoid Transformer diaries.

It is important to emphasize that the resulting stream of the generated output is available online\(^3\). No human post-processing of output is performed.

**Discussion**

In Dostoevsky’s “Notes from the Underground” there is a striking idea about madness as a source of creativity and computational explanation as a killer of artistic magic: “We sometimes choose absolute nonsense because in our foolishness we see in that nonsense the easiest means for attaining a supposed advantage. But when all that is explained and worked out on paper (which is perfectly possible, for it is contemptible and senseless to suppose that some laws of nature man will never understand), then certainly so-called desires will no longer exist.” (Dostoevsky 1984) Paranoid Transformer brings forward an important question about the limitations of the computational approach of creative intelligence, either it belongs to a human or algorithm. This case demonstrates that creative potentiality and generation efficiency could be considerably influenced by such poorly controlled methods as obfuscated supervision and loose interpretation of the generated text.

Creative text generation studies inevitably strive to reveal fundamental cognitive structures that can explain the creative thinking of a human. The suggested framing approach to machine narrative as a narrative of madness brings forward some crucial questions about the nature of creativity and the research perspective on it. In this section, we are going to discuss the notion of creativity that emerges from the results of our studying and reflect on the framing of the text generation algorithm.

What does creativity in terms of text generation mean? Is it a cognitive production of novelty or rather generation of unexpendable meaning? Can we identify any difference in treating human and machine creativity?

In his groundbreaking work (Turing 1950) pinpoints several crucial aspects of intelligence. He states: “If the meaning of the words “machine” and “think” are to be found by

\(^3\)https://github.com/altsoph/paranoid_transformer

examining how they are commonly used it is difficult to escape the conclusion that the meaning and the answer to the question, “Can machines think?” is to be sought in a statistical survey such as a Gallup poll.” This starting argument turned out to be prophetic. It pinpoints the profound challenge for the generative models that use statistical learning principles. Indeed, if creativity is something on the fringe, on the tails of the distribution of outcomes, then it is hard to expect a model that is fitted on the center of distribution to behave in a way that could be subjectively perceived as a creative one. Paranoid Transformer is a result of a conscious attempt to push the model towards a fringe state of proximal madness. This case study serves as a clear illustration that creativity is ontologically opposed to the results of the “Gallup poll.”

Another question that raises discussion around computational creativity deals with a highly speculative notion of self within a generative algorithm. Does a mechanical writer have a notion of self-expression? Considering a wide range of theories of the self (carefully summarized in (Jamwal 2019)), a creative AI generator triggers a new philosophical perspective on this question. As any human self, an artificial self does not develop independently. By following John Locke’s understanding of self as based on memory (Locke 1860), Paranoid Transformer builds itself on memorializing the interactive experience with a human, furthermore, it emotionally inherits to its supervising readers who labelled the training dataset of the supervision system. On the other hand, Figure 4 clearly shows the impact of crypto-anarchic philosophy on the Paranoid Transformers’ notion of self. One can easily interpret the paranoiac utterance of the generator as a doubt about reading and processing unbiased literature.

\[\text{copyrighted protein fiction may be deemed speculative propaganda.}\]

Figure 4: “Copyrighted protein fiction may be deemed speculative propaganda,” – the authors are tempted to proclaim this diary entry the motto of Paranoid Transformer.

According to the cognitive science approach, the construction of self could be revealed in narratives about particular aspects of self (Dennett 2014). In the case of Paranoid Transformer, both visual and verbal self-representation result in nervous and mad narratives that are further enhanced by the reader.

Regarding the problem of framing the study on creative text generators, we cannot avoid the question concerning the novelty of the generated results. Does Paranoid Transformer demonstrate a new result that is different from others in the context of computational creativity? First of all, we can use external validation. At the moment, the Paranoid Transformer‘ book of is prepared to come out of print. Secondly, and probably more importantly here, we can indicate the novelty of the conceptual framing of the study. Since the design and conceptual situatedness influence the novelty of the study (Perišić, Štorga, and Gero 2019), we claim that the
suggested conceptual extension of perceptive horizons of interaction with generative algorithm can solely advocate the novelty of the result.

An important question that deals with framing of the text generation results engages the discussion about the possibility of a chance discovery. In (Ohsawa 2003) lays out three crucial three keys for chance discovery, namely, communication, context shifting, and data mining. (Abe 2011) further enhances these ideas addressing the issue of curation and claiming that a curation is a form of communication. The Paranoid Transformer is a clear case study that is rooted in Ohsawa’s three aspects of chance discovery. Data mining is represented with a choice of data for fine-tuning and the process of fine-tuning itself. Communication is interpreted under Abe’s broader notion of curation as a form of communication. Context shift manifests itself thought the reading the narrative of madness that invests the reader with interpretative freedom and motivates her to pursue the meaning in her own mind though simple, immersive visualization of the systems’ fringe ‘mental state’.

Conclusion

This paper presents a case study of a Paranoid Transformer. It claims that framing the machine-generated narrative as a narrative of madness can intensify the personal experience of the reader. We explicitly address three critical aspects of chance discovery and claim that the resulting system could be perceived as a digital persona in a fringe mental state. The crucial aspect of this perception is the reader, who is motivated to invest meaning into the resulting generative texts. This motivation is built upon several pillars: a challenging visual form, that focuses the reader on the text; obfuscation, that opens the resulting text to broader interpretations; and the implicit narrative of madness, that is achieved with the curation of the dataset for the fine-tuning of the model. Thus we intersect the understanding of computational creativity with the fundamental ideas of receptive theory.

References


Fish, S. E. 1980. Is there a text in this class?: The authority of interpretive communities. Harvard University Press.


Menabrea, L. F., and Lovelace, A. 1842. Sketch of the analytical engine invented by charles babbage.


Comparing Different Methods for Assigning Portuguese Proverbs to News Headlines

Rui Mendes & Hugo Gonçalo Oliveira
CISUC, Department of Informatics Engineering
University of Coimbra, Portugal
rppm@student.dei.uc.pt, hroliv@dei.uc.pt

Abstract
This paper reports on the automatic selection of short-texts for amplifying the range of a given input’s context, e.g. a news headline. Different methods are applied to select a corresponding expression from a list, which should be as semantically related to the input as possible. This study was developed for the Portuguese language, and considered expressions are proverbs, where figurative language is predominant. The set of explored methods includes some based on word overlap, others on static word embeddings, and also on recent contextual embeddings. To compare the explored methods in this subjective scenario, a survey was answered by humans, who rated the value of the selected expressions in terms of relatedness with regard to the input, and the humoristic value that may arise from this selection. The main conclusion was that simpler approaches, which end up selecting expressions that share words with the headline, are more easily related and considered to be funnier than other more elaborate approaches, which are more focused on the context.

Introduction
Topics involving the processing and understanding of natural language are often explored by applications designed for English. The study described in this paper targets the Portuguese language, which presents different challenges and, despite having a large number of speakers, has a much smaller research community. We propose an automatic selector of expressions, e.g. proverbs and sayings, able to choose expressions with regard to the input’s context, e.g. a news headline. This type of expressions includes word-play and is rich in terms of figurative language. Thus, models trained in general language may struggle to interpret them. For this purpose, we test and analyze diverse approaches, and see many possible and different applications. In the domain of journalism, news stories are constantly published in online newspapers, raising the importance of having appealing headlines, which may be achieved by using familiar and figurative expressions that may also imply some form of humor. For instance, Jornal de Léiria’s headline “Burro Velho não aprende línguas, mas mata a fome a quem aparecer" ("Old donkey does not learn languages, but satisfies the customer’s hunger") plays with the proverb “Burro velho não aprende línguas" ("Old donkey does not learn languages") and uses it to increase its appeal, as the news story is about a restaurant named "Burro velho" ("Old donkey"). One of the goals of this study is to automate that process of selection. There are also several tv shows, like the news satire Last Week Tonight, which use short-texts such as proverbs or movie titles to complement scenes. Furthermore, in the domain of chatbots, using proverbs and sayings in the appropriate contexts could make conversations more interesting.

However, for a computer, it is very troublesome to find the underlying meaning beneath the use of figurative language, which is not to be literally interpreted. In spite of that, a computer needs to understand how to find the most similar and humorous relation between different texts, which in this study are represented by an input and a list of expressions. Several approaches can be used in the process of selecting the suitable sayings, starting by the representation of text and its comparison. The set of tested methods comprises simple approaches such as computing the Jaccard similarity or applying the Term Frequency - Inverse Document Frequency (TF-IDF) algorithm, to other more recent approaches such as those based on static Word Embeddings (WEs). Moreover, state-of-the-art methods based on transformers are also analyzed, namely the Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019), which has been getting a lot of attention due to having unexpectedly good results in many Natural Language Processing (NLP) benchmarks. Each of these approaches computes the similarity between two short texts, both with their advantages and disadvantages.

This paper is divided in four different sections, starting by this introduction, which is followed by an overview of background knowledge for better understanding this study. We further present some related works that inspired this paper, followed by the methodology by which this study was developed. Before concluding, we describe how the selected methods were evaluated in the proposed scenario and discuss our interpretation of the evaluation’s results.

Background Knowledge
Before applying methods for natural language processing and understanding, a textual input often needs to be pre-
processed. Each sentence is often submitted to a morphological analysis consisting of: (i) tokenization, the separation of the sentence into words (tokens); (ii) lemmatization, reducing the word to its most basic form; (iii) part-of-speech (PoS) tagging, identifying the word’s class (like nouns or adjectives). Therefore, each word and its derivatives are considered, eliminating words whose contribution to the semantic value of the sentence is limited, such as stopwords, which may be very frequent, e.g. ‘the’ for the English language and ‘a’ for the Portuguese language.

After pre-processing, the meaning of each token is considered, i.e. the semantic value of each word, bearing in mind that different words may have the same or a similar meaning (synonymy), e.g. ‘big’ is semantically similar to ‘large’; or the same word might have more than one distinct meaning (homonymy), e.g. ‘right’ may be a side or may mean ‘correct’.

Considering the semantic value of a word, the distributional hypothesis in linguistics, summarized by the quote “You shall know a word by the company it keeps” (Firth, 1957), assumes that the meaning of a word can be inferred by the context where it is inserted, i.e. the window of words that are near the chosen word.

The concept of similarity is grounded on features shared by two units, and sets their positions in a taxonomy. Semantic similarity measures the similarity of meanings, transmitted by words, and can be estimated by different methods, relying on different representations. A close concept is relatedness or association Budanitsky and Hirst (2006), which considers any other relation that may connect meanings. For instance, dog and cat are similar, but none is similar to bone. Moreover, dog is more related to bone than cat.

However, when it comes to longer sequences of text, the previous are less clear. In this context, Semantic Textual Similarity (STS) (Agirre et al., 2012; Cer et al., 2017) aims at computing the proximity of meaning of fragments of text or sentences. The most simple approaches for STS are based on averaging a semantic representation of each word. Furthermore, it is common to weight each token according to its relevance, using, for instance, the TF-IDF algorithm for this purpose.

### Traditional STS Methods

In the following paragraphs, we present the three simplest STS methods used in this study, considered traditional due to their usage before the introduction of word embeddings (WEs) and other more recent methods.

The Jaccard coefficient computes the similarity between sets as their intersection divided by their reunion. In order to compute sentence similarity using this measure, sentences are represented as sets of tokens. Similarity is then given by the number of shared tokens divided by the total number of distinct tokens in both sentences, possibly after pre-processing.

For the remaining methods, sentences are represented by vectors of numbers, and their similarity given by the cosine of such vectors, differing in how they are computed.

The CountVectorizer method (implementation in Pedregosa et al. (2011)) performs tokenization on a set of textual documents and constructs a vocabulary, enabling the codification of documents in regard to that vocabulary. These sparse vector elements (i.e., with many elements equal to zero) represent the number of times each word appears in the set of documents, and can be used to build co-occurrence matrices, which represent, for each word, the number of times it appears in the context of other words.

TFIDFVectorizer relies on the TF-IDF algorithm to compute the relevance of a word given a certain corpus, based on its frequency. It is usually used as a weighting factor in co-occurrence matrices. Term Frequency is the number of times a word $w$ occurs in a document, while Inverse Document Frequency is the number of documents $w$ appears in. This algorithm is able to reduce the weight of stop words, like prepositions or determiners, that contribute little to the meaning of the text, and increase the weight of words that do not appear very often elsewhere, and are thus more relevant for discriminating between different documents.

### Static Word Embeddings

Representing words in vectors of real numbers, also called word embeddings (WEs), provides a friendly way of computing word similarity or using words as features in a machine learning framework, creating a semantic vector space. Such vectors are often learned from text, the more, the better, as they will better generalize word meanings. However, these models are limited to a single representation for each word, which means that multiple meanings of the same word are compressed into a single vector. WEs also present themselves as a solution to compress sparse vectors resulting from other techniques into dense vectors. They can also be improved by the utilization of the TF-IDF algorithm, as a way to increase the weight of the most relevant words of a sentence.

Regarding how WEs are learned, in this work, two different models were used:

- *GloVe* (Pennington, Socher, and Manning, 2014) is an unsupervised learning embedding algorithm based on a word co-occurrence matrix of probabilities in a textual corpus, finding relations between words.

- *FastText* (Bojanowski et al., 2017) is an algorithm for obtaining vector representations for words using a neural network, also considering character sequences, which may improve the processing of languages with a more complex morphology. It can use Continuous Bag-of-Words (CBOW), which uses the context to predict a word in its middle, or Skip-gram, which uses the distributed representation of a given word to predict the context.

### BERT

A transformer (Vaswani et al., 2017) is a neural network architecture that converts input sequences in output sequences. Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) applies attention and recurrence mechanisms to gather information about the relevant context of a given word and then encode that context in a rich vector that smartly represents the word. Furthermore, a BERT model can be fine-tuned for a specific task,
i.e. it can be specialized in the intended type of text, without having to be trained from scratch.

The self-attention algorithm in the transformer allows for the modelation of many downstream tasks, changing the adequate inputs and outputs. Each word of the input sequence has certain values due to its relations to other words, e.g. “The dog ate its food.”, where “its” is related to “the dog”. This algorithm divides the input sequence, and, for each word $w$, calculates the scores of all words in relation to $w$. The output of the calculation is the sum of all returned vectors created in order to $w$, which is then passed as input for a feed-forward network. BERT has its own way of encoding tokens, starting with its WordPiece tokenization, which may even divide tokens into sub-tokens, e.g. walking or walker become walk@@ing and walk@@er. Even if the model does not know how to deal with the word walking, it probably does know other words that have walk@@er in common, as it will appear more often. Because of this, it only needs the computation of the sentence vector, calculated through the average of the vectors of its tokens. It is also possible to use BERT to directly encode each sentence, instead of a token at a time, which may be an approach to be investigated in the future.

**Related work**

The development of automatic approaches for amplifying the range of a given story through creative artefacts is not new. In this context, systems have been proposed for generating poetry inspired by news stories (Colton, Goodwin, and Veale, 2012; Chrismartin and Manurung, 2015), metaphors according to the current news (Veale, Chen, and Li, 2017); new creative headlines, by resorting to figurative language (Alnajjar, Leppänen, and Toivonen, 2019), or blending them with well-known expressions (Gatti et al., 2015). Moreover, systems have been developed for simply recommending quotes to be used in dialogues (Ahn et al., 2016), without much concern regarding creativity.

Considering the generation of different types of text, namely poetry, Chrismartin and Manurung (2015) consider the dependency relations in a news story and use them for encoding the intended meaning of a poem. To produce text, they used a mechanism named chart generation, guided by the aforementioned relations.

Veale, Chen, and Li (2017) explored the generation of metaphors in regard to the current news. For selecting metaphors, considering their relation to the news, different techniques were explored, namely Latent Dirichlet Allocation (LDA), Latent Semantic Analysis (LSA) and Word2Vec. They were used for constructing a joint vector space that merged a news corpus with a metaphor corpus, thus facilitating their comparisons through the computation of the cosine similarity. After pairing headlines and metaphors, the authors crowd-sourced in order to evaluate the different pairing models in three dimensions: comprehensibility, aptness and influence of the metaphor on the reader’s interpretation of the headline.

Alnajjar, Leppänen, and Toivonen (2019) adapted an automated journalism system that generates news headlines in English and Finnish to increase their creativity. In order to increase catchiness, headlines were enriched with the inclusion of a suitable known expression (e.g. movie title), or with the injection of figurative language, like similes and metaphors, according to the context of the news story. This was achieved with the computation of semantic similarity and by classifying the prosody of the phrase and the headline, which “is evaluated to increase the catchiness of the result”. For the news headline “Biggest vote gains for The Green League in Kuopio", the first approach generates the expression “Alls well that ends well: Biggest vote gains for The Green League in Kuopio”, while the figurative approach returned “Biggest vote gains for The Green League – the lovely god – in Kuopio”.

Gatti et al. (2015) developed a method of generating catchy news headlines using well-known expressions that may also be used as a creativity boosting application. They start with sentences from a corpus of clichés, movie and song titles, or slogans. Then, for the creation of a sentence vector, they sum the sparse vectors representing the occurrences of the words of the sentence, excluding stop words. For each slogan, the most similar headlines were selected, according to the cosine similarity. Afterwards, the authors were able to classify the relations between words based on a dependency treebank, and select the keywords, from the selected news headlines, which were able to replace a word $w$ in the slogan, considering its PoS. When successful, this substitution is ranked together with the other successful substitutions, considering the mean of its similarity and dependency scores. The candidate with the highest rank is selected and presented. Diverse outputs were returned. A example, for the headline “Wood: Time for Wales to step up”, based on the slogan “Unleash the power of the majority” is “Unleash the power of the sun”.

Ahn et al. (2016) proposed a system that recommends known quotes or expressions, given the features of the received short-line text, like dialogues and writing. In this work, there is a clear separation in the definition of context, between pre-context, i.e. texts before the quote, and post-context, i.e. texts next to the quote, within the capabilities of the selected window. This research presents five methods for quote recommendation:

1. Matching Granularity Adjustment: measures the importance of a set of contexts to a query.
2. Random Forest: tree based classification algorithm, chosen due to its resilience to overfitting and tendency to exhibit low variance and bias.
3. Convolutional Neural Network: searches for the best “n-gram features in a given context by learning the parameters of fixed size filters for each n-gram”.
4. Recurrent Neural Network with LSTM: consists of three parts (forget, input, output) to teach the networks long-term dependencies without loss of information.
5. Rank Aggregation: groups the individual results of multiple methods to create a precise ranking of quotes.

Even though humour is not the main target of this work, some results may produce a humouristic effect. On the other hand, still in the scope of Computational Creativity, there
are systems explicitly focused on the generation of humour, e.g. by replacing some words in a given text by others with the same form, context and topic, including taboo meanings (Valitutti et al., 2016), which are effective methods to increase the average funniness of a sentence.

Concerning the Portuguese language, related systems have focused on news headlines for the generation of memes, using a popular image macro and a text related to the news (Gonçalo Oliveira, Costa, and Pinto, 2016). Receiving as input a news headline, the previous system selects an image, from a predefined set, which is considered related to the input and adapts the text according to the stylistic rules that the meme textual content must abide, so that the combination of text and image produce humoristic content. Another example did not use news but Twitter trends, in this case as an inspiration for automatically generated poetry (Gonçalo Oliveira, 2017).

Methodology

We recall that the main goal of this work is to compare methods for assigning related Portuguese proverbs, automatically, to Portuguese news headlines. For this purpose, the main requirements are: (i) a collection of news headlines (in the future, these can be retrieved in real-time); (ii) a collection of expressions; (iii) an assignment method.

The development of this system was written in Python 3.6, with aid from various Python adapted libraries, both for text related operations, but for statistical purposes as well. The first step was to gather a good collection of data on the chosen context, accomplished by gathering news from the News API sources, found in Portuguese newspapers’ online editions. The News API allows a client to get a maximum of one hundred current news on given keyword(s), returning an object with news whose titles are similar to the keyword(s). For this work, the news were reduced to their headline, in order to work with texts that do not differ too much in length, and were chosen both by date and by keyword, i.e. the API returned all the news related to the keywords ‘clima’ (‘climate’), ‘ambiente’ (‘environment’) and ‘aquecimento global’ (‘global warming’), posted on the three months previous to the API call (February 2020).

Alongside the news, it is important to have a large enough corpus of proverbs. In this case, we used a corpus of 1,617 Portuguese proverbs, obtained from project Natura of Universidade do Minho. Once we had the headlines and the proverbs, we decided on a range of methods for computing sentence similarity, to be later compared, namely:

- Jaccard Similarity
- TfidfVectorizer
- CountVectorizer

- WEs generated by GloVe
- WEs generated by GloVe plus the weighing of the TF-IDF algorithm
- WEs generated by FastText (FT)
- WEs generated by FastText plus the weighing of the TF-IDF algorithm
- Bidirectional Encoder Representations from Transformers (BERT)

For both the Count Vectorizer and the TF-IDF Vectorizer method, Python’s scikit-learn library (Pedregosa et al., 2011) was used to perform the calculations necessary to encode the words and their occurrences into matrices. These matrices are then used to compute the similarity between the input and the list of sayings, through the cosine_similarity method from the mentioned library.

Four variations of WEs-based methods were tested. Using Python’s Gensim library, two different models were used: Glove and FastText (using CBOW). Both were models pre-trained for Portuguese, with 300-dimension vectors: GloVe was obtained from the NILC repository of Portuguese word embeddings (Hartmann et al., 2017), and the FastText model from the FastText repository, where models are available for several languages, including Portuguese. Variations of sentence representations using the previous relied on TF-IDF for weighting word vectors. Moreover, to represent text with WEs, the short-text is submitted to a pre-process that includes tokenization and turning tokens to lowercase, accepting only those that are present in the model’s vocabulary. The sentence vector is then computed from the average of its tokens’ WEs. To finalize, we used Gensim’s method for computing the cosine similarity – existent for any of the mentioned models – is used between the input’s and proverb’s sentence vector.

For the application of BERT, a pre-trained multilingual model was used, available by Google and covering 104 languages, including Portuguese: BERT-Base, Multilingual Cased. For this method, a Python library named bert-as-service was used, requiring to run a BERT server with the mentioned model, which is then called by the client, i.e. the system. Due to BERT’s specific vector encoding, the cosine similarity is calculated with the help of Python’s NumPy library, instead of the previously mentioned methods.

For most of the methods, both headlines and proverbs were first pre-processed with the NLPyPort package (Ferreira, Gonçalo Oliveira, and Rodrigues, 2019), a layer on top of the Natural Language Toolkit (NLTK) (Loper and Bird, 2002) tackling Portuguese, specifically. This enabled the linguistic pre-processing, namely with tokenization and PoS tagging, and was essential for further application of the sim-
ilarity methods, as seen above. Only BERT does not need this pre-process.

Afterwards, the similarity of each headline with each proverb is computed and the proverb with the highest similarity score is used. In this work, this is done for the eight tested methods, all relying in the computation of the cosine similarity between the vectors representing each sentence, which were computed by the average of the vectors of the sentence tokens. Following the computation of similarity, the proverb with the highest similarity score is selected to represent the correspondent approach, e.g. for the headline “Produção de combustíveis fósseis cresce 50% acima do necessário para travar aquecimento global” (“Fossil fuel production grows 50% above what is needed to curb global warming”), a good choice, in our opinion, would be “Quem dá e torna a tirar ao inferno vai parar” (“Those who give but take back, end up in hell”).

**Evaluation**

Evaluating headline-proverb pairings is a subjective task. Therefore, in order to assess the results of each method, it was necessary to resort to human opinions, more precisely 24 volunteers who were asked to answer a survey. They were grouped into six teams of four people, each assigned to ten news headlines from a total of 60, summing a total of 240 different evaluations.

In the survey, created with Google Forms¹¹, for each of the ten assigned news headlines, the volunteer judge had to classify the selected expressions with regard to two aspects: (i) relatedness, which classifies the semantic proximity between the expression and the input; (ii) funniness, a classification for the humoristic value of the expression considering the input’s context, but not limited by it, as a judge may find an expression funny by itself. Although we could think of other relevant aspects, we focused on the previous two. There was no need to validate the syntax (every expression was unaltered, only selected), and the aspect of originality is more pertinent for other endeavours, namely regarding works in text generation. For the scope of this evaluation, judges did not have to justify the score they gave each expression, as it could influence their opinion.

As an example, for the headline “Emissores atmosféricos aumentaram em 2017” (“Atmospheric emissions raised in 2017”), the volunteers were asked “Como avaliaria a relação entre os provérbios e a notícia?” (“How would you rate the relation between the proverbs and the news title?”). Below these questions, they would see the list of selected proverbs in a random order, with no repetitions, and rate each one according to a 4-point scale: Not related (1); Remotely related (2); Considerably related (3); Extremely related (4). Afterwards, regarding the funniness of each proverb, they were asked “Relacionando com o título, qual engraçado é cada provérbio?” (“In relation to the headline, how funny is each proverb?”), to which the answers were also rated on a 4-point scale: Not funny (1); Remotely funny (2); Considerably funny (3); Extremely funny (4).

Table 1 reveals the results of this evaluation, respectively for the relation between proverb and headline and its funniness. Besides the distribution of scores for each evaluated aspect, it shows the median (Md), the means (µ) and standard deviation (σ), which could work as an overall score. To measure inter-rater agreement, we used Fleiss’ kappa (Fleiss, 1971), a coefficient similar to Cohen’s Kappa (Landis and Koch, 1977), that also considers the possibility of judges agreeing by chance. The difference is that Cohen’s Kappa only works for two judges, while Fleiss’ applies to more than two. Since we had six different surveys, each answered by four judges, Fleiss’ kappa was computed for each survey and each validated aspect. For relatedness, it was 0.094, and for funniness 0.054. Following the standard guidelines for interpreting these values, there is just a slight agreement in terms of relatedness and funniness, highlighting the subjectivity of this kind of evaluations.

As previously stated, the proverbs selected by each method were evaluated regarding their relatedness with the headline and whether they had humoristic value or not. Considering Table 1, the majority of the results were only satisfactory, both for relatedness and funniness. Most of the proverbs only had a slight relation to their headlines, with every method scoring at least 3 for more than 30% of the times. The idea that mixing proverbs with news headlines may give the headline a humorous touch is supported by similar results, with most methods scoring at least 3 in over 40% of the evaluations.

It is also possible to state that theoretically simpler methods achieved the best scores in the selection of proverbs for given headlines. Particularly, the simplest one, the Jaccard similarity, which is the only method whose proportion of top-scores (4) is higher than 20% in both relatedness and funniness. With regard to this realization, it is possible to argue that Jaccard has the highest scores due to its selection of proverbs with the most words in common with the headline, thus making it easier for people to make a quick and immediate connection between them. The same can be said for the TF-IDF and the Count Vectorizer, which follow Jaccard in the ranking of best relatedness scores.

A curious fact is that methods based in more recent semantic models, namely those using WEs, both with GloVe and FastText, and BERT, achieved lower scores. Even though they selected proverbs whose meaning may not be too far from the headline, their relation is, perhaps, not as pinpointed or clear as in the simpler methods. Considering the methods that rely on WEs, it is possible to conclude that those based on GloVe have higher scores than those using FastText, particularly in terms of funniness, where GloVe is the only method with over 40% answers scored with at least 3.

Table 2 is an indication of the methods with higher global success in terms of the number of times they were selected, by considering the proverbs which, based on the average of their four human opinions, scored at least 3.5 points in terms of relatedness. In the lead is again the simplest algorithm, Jaccard similarity, followed closely by TF-IDF Vectorizer.

The table also presents the average of shared tokens between the selected expression and its correspondent news

---

¹¹https://www.google.com/forms/

---
headlines, for those top-related proverbs and also for all selected proverbs, for comparison purposes. This statistic is higher for the Count Vectorizer and Jaccard, two of the approaches with the most selections with at least 3.5 points of score in terms of relatedness. The more complex and recent approaches did not have as many successful selections, but even these selections had an average of at least two common words with the headline. However, when compared to the average of intersections for the total of selections, differences highlight that more related selections indeed share more tokens with the headline. Therefore, we may argue that people are quicker to relate two sentences that share many words, in regards to their full semantic value chosen by the approaches that are more up-to-date.

Despite having lower scores, BERT was able to make a selection that was scored with an average of 4 points. This was the third example in Table 3, where the chosen proverb’s meaning and urgency clearly applies and is related to the headline. Moreover, BERT had one of the highest proportions of selections with maximum funniness. This might have been due to the surprising effect, or just by chance, as the means suggest.

Another good example of a successful proverb selection, regarding the relation between title and proverb, is the first example in Table 3, which scored 4 for all its judges. Using the TF-IDF Vectorizer, the system selected a proverb whose meaning may correlate with the title’s meaning, as they share the word “lixo” (“trash”), for example.

With concern to the best funniness related results, as seen in the fourth example of Table 3, the selected proverb had the average score of 4. It was selected by Jaccard similarity and uses taboo words, close to slang, which may be the reason for its high score. Taboo words “are often used to produce humor effects” (Valitutti et al., 2016).

In opposition to the highest-scored examples, two of the selections with the lowest score in terms of both relatedness and funniness are depicted in Table 4. Both of them were obtained with WEs, even though the first used the GloVe model and the second the FastText model. The first example tries to make use of the existence of the integer present in the headline, selecting an expression that allures to the commutative property of the multiplication of two real numbers, whose order does not change the end product. In the second example, it is difficult to grasp the scope of similarity between these two sentences, so much as to find their relation funny.

### Conclusions

This study targeted a task of automatic text recommendation, in this case, Portuguese proverbs to news headlines. For this purpose, different semantic representation techniques were tested in this domain for computing the STS between proverbs in a corpus and headlines. To some extent, an application including some of those methods could be useful for writers and journalists, i.e., for making their news more appealing. The produced results were satisfactory, as most of the time people were able to establish a relation between the selected expression and the correspondent headline, and even often find it potentially funny.

Given some thought, the obtained results are particularly interesting, as they indicate that the proverbs sharing the same words with the headline, namely those chosen by simpler methods such as the Jaccard similarity, are more easily related to the headlines. On the other hand, proverbs selected by methods based on state-of-the-art models did not score as high amongst our volunteers. The former methods are based exclusively on the surface text, while the others rely on a deeper semantic representation of words. Though, regardless on whether they could select stronger related proverbs or not, according to our judges, they do not suit well this scenario, where figurative language is predominant. On this, Veale (2015) claims that human readers may suffer from a placebo effect, as they “fill these containers with their own meanings, to see meaning in the outputs of generative systems where none was ever intended”.

<table>
<thead>
<tr>
<th>Method</th>
<th>Relatedness</th>
<th>Funniness</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>μ ± σ Md 1(%) 2(%) 3(%) 4(%)</td>
<td>μ ± σ Md 1(%) 2(%) 3(%) 4(%)</td>
</tr>
<tr>
<td>Jaccard</td>
<td>2.4 ± 1.15 28.8 24.2 22.0 25.0</td>
<td>2.5 ± 1.06 22.0 28.8 27.5 21.7</td>
</tr>
<tr>
<td>Count Vectorizer</td>
<td>2.2 ± 1.06 35.1 26.3 24.6 14.0</td>
<td>2.3 ± 1.04 25.4 33.9 23.7 17.0</td>
</tr>
<tr>
<td>TFIDF Vectorizer</td>
<td>2.3 ± 1.09 32.5 24.6 25.9 17.0</td>
<td>2.3 ± 1.01 27.5 30.5 28.8 13.2</td>
</tr>
<tr>
<td>GloVe</td>
<td>2.2 ± 1.03 33.5 29.2 24.6 12.7</td>
<td>2.2 ± 1.02 32.6 23.7 33.1 10.6</td>
</tr>
<tr>
<td>GloVe+TFIDF</td>
<td>2.1 ± 1.05 37.7 26.3 23.7 12.3</td>
<td>2.2 ± 0.99 31.0 31.7 26.3 11.0</td>
</tr>
<tr>
<td>FT</td>
<td>2.0 ± 1.03 40.7 26.7 21.6 11.0</td>
<td>2.0 ± 0.97 39.4 28.4 25.0 7.2</td>
</tr>
<tr>
<td>FT+TFIDF</td>
<td>2.1 ± 1.03 39.0 25.4 25.0 10.6</td>
<td>2.2 ± 1.05 34.3 26.7 25.4 13.6</td>
</tr>
<tr>
<td>BERT</td>
<td>2.1 ± 1.04 41.1 23.7 24.6 10.6</td>
<td>2.2 ± 1.11 37.2 26.7 19.1 17.0</td>
</tr>
</tbody>
</table>

Table 1: Evaluation of proverb assignment to headlines.

<table>
<thead>
<tr>
<th>Method</th>
<th>Top</th>
<th>μ(Intersection)±σ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jaccard</td>
<td>7</td>
<td>2.6 ± 1.3</td>
</tr>
<tr>
<td>Count Vectorizer</td>
<td>3</td>
<td>3.7 ± 0.6</td>
</tr>
<tr>
<td>TFIDF Vectorizer</td>
<td>6</td>
<td>2.3 ± 1.0</td>
</tr>
<tr>
<td>GloVe</td>
<td>2</td>
<td>3.5 ± 2.1</td>
</tr>
<tr>
<td>GloVe + TFIDF</td>
<td>1</td>
<td>2.0 ± 0.0</td>
</tr>
<tr>
<td>FT</td>
<td>2</td>
<td>2.5 ± 0.7</td>
</tr>
<tr>
<td>FT + TFIDF</td>
<td>1</td>
<td>3.0 ± 0.0</td>
</tr>
<tr>
<td>BERT</td>
<td>1</td>
<td>3.0 ± 0.0</td>
</tr>
</tbody>
</table>

Table 2: Number of selections with more than 3.5 average relatedness score (Top), and the average of token intersections between the proverb and the news headline, both for the top selections and for all selections by each method.
In terms of future endeavours, a possible research direction would be to test supervised approaches for Semantic Textual Similarity (STS) in this scenario, including, but not limited to, fine-tuning BERT. Although there is no gold data with headline-proverb similarity available, we may try using collections for STS in Portuguese (Fonseca et al., 2016) for training such a model. We may also test the same approach with the recently available BERT model trained for Portuguese (Souza, Nogueira, and Lotufo, 2019), or try learning a model from Twitter or Reddit conversations where proverbs are used. This would require looking for tweets using any of the proverbs in the knowledge base and, if there is one, retrieve also the preceding publication(s).

This work was also key in the development of the creative system TECo: Texto Em Contexto (Mendes and Gonçalo Oliveira, 2020), in English Text in Context, which selects and adapts textual expressions based on a given textual input, also in Portuguese Adaptation methods result in new expressions, thus more novel, that still resemble original sayings, with increased relatedness, even when there are no related sayings available. The methods described here are responsible for selecting an initial set of expressions to adapt, and finally select the resulting expression to exhibit, out of several produced.

Acknowledgments

We to thank all the people who answered the survey, without whom the presented results would not be possible.

This work was partially supported by FCT’s INCoDe 2030 initiative, in the scope of the demonstration project AIA, “Apoio Inteligente a Empreendedores (chatbots)”.

References


Alnajjar, K.; Leppänen, L.; and Toivonen, H. 2019. No time like the present: Methods for generating colourful and factual multilingual news headlines. In Proceedings of 10th International Conference on Computational Cre-
activity (ICCC), 258–265. Association for Computational Creativity.


A Pragmatics-based Model for Narrative Dialogue Generation

Andreea-Oana Petac¹, Anne-Gwenn Bosser¹, Fred Charles², Pierre De Loor¹, Marc Cavazza³

¹École Nationale d’Ingénieurs de Brest, France
²Bournemouth University, Faculty of Science and Technology, United Kingdom
³University of Greenwich, London, United Kingdom

petac@enib.fr, bosser@enib.fr, fcharles@bournemouth.ac.uk, deloor@enib.fr, m.cavazza@greenwich.ac.uk

Abstract

We describe a method and a proof of concept which allow the generation of rich and engaging dialogues between virtual characters from a formalised plot description. The structure of the dialogue generated borrows from inferential pragmatics, following the Geneva Model of discourse analysis, in order to provide realistic interaction between characters in the narrative. At a higher level, this discourse is organised following heuristics borrowed from narratology theory in order to elicit emotions linked to dramatic tension and thus favour narrative engagement. Besides enriching narrative generation systems embedded within simulation applications, our work also has the potential to be adapted to support engaging interactive dialogues between users and virtual conversational agents in narrative systems.

Related Works

Interactive storytelling systems have provided a mechanism to generate narratives based on computational models, and allowing the plot to be instantiated either through visuals or text (Cavazza, Charles, and Mead 2002). Furthermore, generating dialogues in interactive storytelling has been an important means of delivery of the story discourse. Dialogues are produced based on story formalism, through rule-based mechanisms. The development of dialogues as part of storytelling systems has been tackled over the years by several researchers, such as (Cavazza and Charles 2005; Onate, Mendez, and Gervas 2019), or in the seminal Façade system (Mateas and Stern 2004) which is one of the early attempts at relating the principles of narrative functions to dialogue acts.

Whilst narrative discourse often tends to be generated through a monologue, there have been several attempts at generating dialogues to reflect their richer nature. (Piwek and Stoyanchev 2011) created a system able to transform a monologue into a two-participant dialogue. (Bowden et al. 2016) notes that information provided in the form of a dialogue is more engaging than when provided in monologue form. They present algorithms for converting a deep representation of a story into dialogue-based storytelling that can vary aspects of the conversation.

Introduction

Research in interactive storytelling has in the last 20 years produced a number of prototype systems allowing to engage their users in simulations where virtual characters would play out a story influenced through user interaction. These systems rely on real-time story generation mechanisms in order to generate story events according to high level narrative goals, or repair the unfolding story with regards to user interaction, often staging virtual characters’ interaction and scripted dialogue as the lowest level narrative actions, which therefore sets a limit to the story adaptation and to the richness and variety of characters’ dialogues. In this paper, we describe an approach and a proof of concept which would allow to overcome this limitation thanks to the generation of rich and engaging dialogues between virtual characters from a formalised plot description. The dialogues generated by the system deliver narrative content through the conversation between these agents.

Our approach builds upon theories from two disciplines. First, we rely on dialogue structures, as defined in the Geneva Model of Discourse Analysis (Filliettaz and Roulet 2002; Moeschler 2002), used as the building blocks for modelling conversations. Second, we exploit the discourse organisation from Baroni’s work (2007), generating curiosity as the effect of a temporary discourse incompleteness of the narrative action.

Dialogue Structures and their Relations

The Geneva School of Linguistics (Filliettaz and Roulet 2002) proposed a modular approach to discourse analysis. Discourse is composed of a finite set of components combined in a hierarchical representation. To retain the richness of the discourse, we consider a multi-component approach. Based on (Filliettaz and Roulet 2002), we define two theoretical categories needed to transform our monologue into a dialogue. The two main questions to answer are: i) how do we organise the discourse and ii) how do we describe the links between the monologue statements that are essential in the logical and causal narrative of the generated discourse.

To address the first question, we propose using a set of discourse structures initially described in (Moeschler 2002) in order to analyse monologues and thus, providing us with a strong theoretical basis; we use these structures to generate dialogues, as further described herein. If no specific formalisation is defined for linking these structures, then the re-
Integrating Narrative Tension

Classical narratology has described means to achieve narrative tension to engage readers in the unfolding of the story. Russian formalist Tomachevski, in an essay published in 1925 (translated in French in Tzvetan 1965), remarked the discrepancy between narrative events and their presentation to the reader through the discourse. This distinction has now been widely accepted in contemporary narratology. We use plot, story and discourse to refer respectively to the logical organisation of the narrative, the elements of the narrative, and the way they are presented to the reader, a terminology which is common in the Computational Narratives community. Tomachevski remarked how narrative discourses are constructing secrets, voluntarily maintaining readers in the ignorance of details which are necessary to understand the story, thus triggering curiosity. This aspect of narrative tension has also been described by Barthes (1970): the hermeneutic code introduces mysteries at the beginning of the text, in the form of an unexplained event that will be solved later in the text. Such constructions are typical, for instance, of whodunnit mystery novels and movies, which engage their audience using the intellectual curiosity for understanding an emotionally charged event which is exposed early in the narrative.

Sternberg (2003) described the centrality of curiosity, suspense and surprise in narratives. These concepts have been further honed by Baroni (2007), who clarified how narrative tension can be either heuristic or dramatic, both having to rely on the organisation of discourse but the latter relying more heavily on interpretation of the reader. Naturally, altering the interventions within the narrative will lead to different effects. Building on the work of (Sternberg 2003), (Baroni 2007) has described how the structure of discourse can elicit various emotional effects, resulting in the building up of dramatic tension, allowing to emphasise narrative events and to emotionally engage the audience. We show in this paper how the heuristic tension described by Baroni (2007) can be simply structurally built in the generated dialogue, by maintaining the audience in ignorance of narrative events which is essential for narrative comprehension. Furthermore, (Wu, Young, and Christie 2016) focus on deciding the most appropriate places to insert story events as flashback effects (i.e. curiosity effects), and the impact these effects have on the readers.

Suspense in narrative generation plays a key part in terms of the affective responses elicited to induce reactions by the user. Reactions in response to this type of engaging interactive stories are positively related to enjoyment, having a significant impact on the audience’s immersion and suspension of disbelief (Delatorre et al. 2018). Several interactive storytelling systems, such as Suspenser (Cheong and Young 2014), have ensured to incorporate this principle at the very heart of the generation process to enhance the quality of the experience generated for the users.

We propose here to manipulate the dialogues using such narrative constructions in order to generate a compelling discourse. In the remainder of the paper, the components of this proposed approach (Fig. 1) are described, along with our own taxonomy combining these constituents in creating complex discourse structures.

Generating Structured Dialogues

A dialogue model is based on a discourse structured in a predetermined sequence of types of utterances. To represent the hierarchical structures in which the dialogue is organised, we need to account for the rules of formation proposed by (Filliettaz and Roulet 2002; Moeschler 1989; 1translated from the French “tension heuristique”
2002). The dialogue is represented as a hierarchical structure on which the dialogic units (Exchanges, Moves and Dialogue Acts) are mapped. A dialogue model consists of descriptive units (i.e. the structure) and the relations that connects these structures to each other (Pernel 1994).

- Exchanges (E) describe the main structure component of the dialogue. The usage of exchanges marks discourse segments according to the considered topic.
- Moves (M) relate to the exchange depending on topic similarity. Exchanges can be embedded within Moves to represent a particular view within the same topic.
- dialogue Acts (A) are associated to a communicative goal.

To formalise our dialogue structures, we propose to define various types of categories of representations. We base our formalisation of these structures upon the rules defined in (Moeschler 2002). To computationally generate dialogues, we first need to express them in a concise form. We choose to represent these rules in Backus Naur Form (BNF) notation. We use the notation \( n \times X \), whereby at least \( n \) units of type \( X \) (default value is 0), \( U \) is the dialogue unit, \( E \) is the exchange, \( M \) is the move and \( A \) is the dialogue act. The notation :: is read as it; and the notation | is read as or. The first rule below is read as a dialogue unit is either at least one dialogue Act, at least one Move, or at least one Exchange.

\[
U := 1^*A|1^*M|1^*E \\
E := 2^*M \\
M := *U 1^*(M/A) *U
\]

Text Relation Markers

Roulet (2006) specifies that text relations (TR) are not only defined between text constituents (Blakemore 1992; Van Dijk 1979), but are defined as the relation between the discourse structure and the text constituents. As a result, their description depends on their hierarchical structure and on the occurrence or insertion of a specific text relation marker (TRM; (Roulet 2006)) of a particular category.

Text relations (TRs) are classified into ten different levels, some characterising what is happening at the Exchange level, while others are specific to the Move level. An Exchange can be characterised as either initiative or reactive. Move levels have eight types of relations: topicalization, counter-argument, preliminary, commentary, argument, reformulation, succession and clarification. Roulet (2006) describes how to use each relation based on the analysis of a narrative. However, we here make use of these for the purpose of generation, thus we now need to define specific templates for these.

Integrating Text Relations

Through the use of TR and their respective characteristics, we consider one other level of constraint that helps group together the relations into particular structures. We consider that TRMs are divided in two categories due to some relations describing a major step in the completion of the dialogue, hence having a higher importance in the Exchange pairs. Also, based on their structural representation and characteristics, some relations are dependent on others. For example, a Preliminary relation is part of the Argumentation relation – no matter whether the Preliminary is either a single statement or made out of a Conjunction relation, but by its definition, it requires some type of a main constituent afterwards – in consequence, being followed by an Argument type statement.

We now consider the connective elements as divided in two categories: logical conjunction and TRMs, as shown in Fig. 2, where \( S \) represents the statements.

Category: Logical Conjunction

It connects two subordinate statements linked by a conjunction “and” (Fig. 3). Its main purpose is to add strong connected information to the discourse. Based on the defined taxonomy, the relation is thus commutative.

Figure 3: In the case of logical conjunctions, utterances are connected to the same Move, the speech acts being linked through a “and” conjunction.

Despite being used as an additive relation, it provides various response strategies:

- provide continuation for effect The second interlocutor finishes the actual relation, the second clause together with “and" being generated from their point of view.
- request details The second interlocutor expresses an explicit request for additional information.
- request importance The second interlocutor requests the importance of a statement for future references (e.g.
“Why do I have to know about this?”). With regards to the
effect, there is an importance for what was expressed.

- **backchannel** The backchannel interventions (e.g., “Yeah /
Right/ Uh-huh”) expressed by the second interlocutor.
   Concerning the effect, the first interlocutor continues with
   their story, using any type of relations.

**Category: Argument**

It expresses a statement describing a point of view within the
same topic, and is either preceded or followed by a prelimi-
nary move; the relation is thus commutative. A preliminary
move can be comprised of only one preliminary statement
or multiple statements grouped by the logical conjunction
“and” (Fig. 4). This relation is indicated by the following
TRMs, Argument.TM: because (of), since, as, like, even,
moreover, if, then, therefore, for that, so that, at least.

![Figure 4: Structural effects for the Preliminary statement S_A
and the Argument statement S_B (this relation is commuta-
tive, i.e. S_B can be used as a Preliminary whereas S_A as the
Argument).](image)

We define the strategies for generating a dialogue compris-
ing of M.ARG (and thus Argument.TM), from the sec-
ond interlocutor’s point of view:

- **ask argument for preliminary** The second interlocutor ei-
ther asks for an argument that sustains the preliminary, or
   they reply with the actual argument. One way to ask for
   the argument is by using the preliminary statement itself,
   explicitly asking for a reason. This type of reply is always
   placed after the first move of the relation being generated.
   Regarding its effect, an argument is given to support the
   claim.

- **interject** The second interlocutor replies with an interjec-
tion or exclamation. This kind of move allows to present
   how the second interlocutor feels about what has been
   said.

**Category: Counterargument**

It expresses a statement describing a point of view in opposi-
tion to what was previously generated, within the same
topic. It is either preceded or followed by a preliminary
move; the relation is thus commutative. It is indicated by
the following TRMs, COUNTARG.TM: although, whatever ...
that, whatever, even if, but, nevertheless, however, even
though, only, being true that, however many/much, despite.

We define the strategies for constructing a dialogue compris-
ing of M.COUNTARG, from the second interlocutor’s side:

- **give reasoning countarg** The second interlocutor inter-
venes by adding a possible reasoning (either a cause or an
   effect). From the structure’s point of view, the state-
ment joined with the TRM completes the counter argu-
ment exchange. Regarding the effect, an opposing claim
is included into the discussion.

- **interject** The second interlocutor replies with an interjec-
tion or exclamation. This kind of intervention shows how
   the second interlocutor feels about what has been said
   (e.g. the received counterargument is not the one they
   were expecting).

**Category: Topicalization**

It highlights the focus of the next part of the discourse, i.e.
an emphasis towards a word, a connection, or a reference. It
is represented by a statement linked to the discourse by the
topicalization TRM (Fig. 5). This relation can appear as part
of another relation, or it can be independent from other con-
stituents. Regarding the structural effect, it opens up a new
speech act inside the current move. Roulet (2006) suggests
the following TRMs for this category, TOPIC.TM: as for,
regarding, concerning, with respect to, as regards.

![Figure 5: Structural effect for a Topicalization move; this
move contains the elements of TOPIC.TM and the actual
narrative statement, S.](image)

**Category: Commentary**

It is either represented by one statement, or via the Conjunc-
tion relation (Fig. 6). Although this category does not have
a specific list of TRMs, Roulet (2006) defines it as always
needing to follow the main constituent. Thus, the goal of this
relation is to further describe the main constituent by hav-
ing a more in-depth description of the previous statement’s
topic, and also providing the other interlocutor with a con-
tinuation of the story. We characterise this relation as always
being dependent on another relation with a higher meaning
in the discourse; in other words, it will always appear as a
constituent of other relations.

![Figure 6: The Commentary relation can either be expressed
by a single statement S_A, either through a logical conjunc-
tion move (S_B and S_C).](image)

**Category: Preliminary**

It marks the beginning of a new topic. Similar to the Com-
mentary relation, with the only difference that the main con-
stituent must follow after the Preliminary move. There are
no specific components for this category’s TRMs.

**Category: Reformulation**

As shown in Fig. 2, this relation is order dependent. Re-
garding the structural representation, the main constituent
of the Reformulation is preceded by a Topicalization statement (M_TOPIC) and followed by a Commentary move (M_COMM) that can either be expressed as a single statement, or as two clauses linked together by the logical conjunction M_CONJ (Fig. 7). In discourse theory (Roulet 2006), the Reformulation’s main constituent is indicated by the following TRMs, Reform_TM: in fact, basically, in any case, anyway, finally, after all, in short/shortly.

Figure 7: Structural effect for one Reformulation statement $S_D$, preceded by the Topicalization statement $S_A$, and succeeded by a Commentary-type intervention expressed through $S_C$.

We define the strategies for building up a dialogue comprising of M_REFORM (thus, by using Reform_TM):

- requestplanation An intermediate reply that validates the Topicalization statement, marking a request for an explanation or a verification of the intended topic. This reply opens up an exchange inside the current one, pausing the narration process until the first interlocutor intervenes with a Clarification with their turn to close the current exchange. The dialogue will then continue with the Reformulation relation.

- wrap up A reply with the role of wrapping-up an idea/fact that has just been covered during the Commentary relation. It could also be used to check understanding. This type of move is added after the Commentary relation at the current level.

Category: Clarification It must succeed an interrogative intervention. No specific TRM describes this category. It is used during the requestplanation case of the Reformulation strategies of replies.

Category: Succession
It describes further development of the discourse, through a continuation of the statements being told to the second interlocutor; Explicitly shows the progression of the narrative. In discourse theory (Roulet 2006), the Succession relation is indicated by the following TRMs, SUCCESION_TM: then, after that, as soon as. Regarding its structure representation, a new simple move is added into the current dialogic unit.

Curiosity in Dialogue Generation
From the relations and story events, the algorithm constructs a tree structure, similar to the one in Fig. 8. Based on the generated discourse instantiation and the defined generation rules, there exist potentially a large number of tree structures to be generated, however constraints are specified to ensure control over the process. These generation rules are integrated into the relation definitions themselves, resulting in consistent structures while still being valid instances of the original model. The tree structure that we constructed allows for branching new substructures in the tree, such as swapping nodes, insertion, deletion. This functionality allows us to easily describe the templates that the dialogue relations are based on. Based on our definition of the curiosity effect, the order of identified tagged relations is modified, resulting in an altered discourse. Our definition of curiosity translates into moving the last relation in the annotated subtree to the first position. Also, if there is a Topicalization structure in the tagged subtree, which according to the definition of Topicalization, highlights the constituent of the current discourse – this is moved to precede the constituent. The algorithm only affects the subtree annotated by the author, and therefore, the order of the rest of discourse relations is preserved based on the discourse. For illustration in Fig. 1.c, the highlighted constituents are considered to be the beginning of the curiosity driven effect.

Worked Example

The Personal Narrative
We illustrate our approach through a detailed example focusing on personal stories, narrated from a first-person point of view (Fig. 1.a). The personal story for this example is from the point of view of Tahani, one of the main characters in The Good Place TV series2.

“I am Tahani Al-Jamil [S1]. I require no introduction [S2] as everybody knows me [S3] because I am a famous British philanthropist [S4], and I am a model [S5]. I work with the most respected agencies [S6]. Now my parents, they are not British citizens [S7]. My father is from India [S8] and my mum comes from Pakistan [S9]. They had another child [S10] because I was not good enough for them [S11]. Anyway, I have a younger sister, Kamila [S12]. With respect to Kamila, everybody prefers her [S13]. Everybody likes her [S14]. We both grew up in England [S15], and we had the best education [S16].

Regarding my education, I went to Oxford [S17], and I attended Sorbonne [S18]. My sister was having a party [S19]. Although I was her sister [S20], I wasn’t invited to the party [S21], but I went there anyway [S22]. Concerning my sister, she had ordered a huge replica of her made of gold [S23]. Because I was so angry on her [S24], I wanted to destroy the statue [S25]. As soon as I started hitting the statue [S26], it started to be unbalanced [S27], and it fell over me [S28]. Since Kamila’s statue was so heavy [S29], it squished me [S30], then it killed me [S31]. Although I have had an amazing life [S32], the end of my life wasn’t as great [S33]. As for how my death went [S34], it happened quickly [S35]. In fact, it was a sudden event [S36] and it was in front of everybody [S37]. After my death, I got to the Good Place [S38]. ”

Story Representation
Through the narrative representation process (Fig. 1.b), we extract the possible sequences of events according to the de-
defined relations. To represent narrative events, we first identify the statements based on TRMs defined in our model; and through the categories that they belong to, we outline the relations between the connected statements.

Table 1 illustrates the representation of the extracted statements for the second paragraph. These relations obtained from narrative representation construct the plot, currently through the authoring process (Fig. 1.c). We only include the statements included in this example plot that is fed into our preliminary system, obtaining the dialogue in Fig. 8. Story events are selected based on our assumption that, in personal stories, the temporal order of events is the same as the order of discourse.

In the following, by treating the groups of relations within the plot, we explain our rationale based on our model. Fig. 8 provides a complete example of a generated dialogue based on the defined plot. Exchanges and Moves are annotated corresponding to this plot’s relations with labels a to m.

| a | ARG(PRELIM(S3), S4) |
| b | Reform(TOPIC(S7), S10, COMM(S12)) |
| c | ARG(PRELIM(S10), S11) |
| d | Reform(TOPIC(S13), S14, COMM(CONJ(S15, S16))) |
| e | Conj(S17, S18) |
| f | Countarg(PRELIM(S20), Countarg(PRELIM(S21, S22)) |
| g | TOPIC(S23) |
| h | Succession(CONJ(S26, S27)) |
| i | ARG(S29, PRELIM(S30)) |
| j | Succession(CONJ(S30, S31f)) |
| k | Countarg(PRELIM(S32, S33)) |
| l | Reform(TOPIC(S34), S35, COMM(S36)) |
| m | Succession(S38) |

Table 1: Sequence of Relations. (Labels on the left hand side match labels in Fig. 8).

**Conjunction** In the current example, in Fig. 8.e, the move consists of the Conjunction relation itself; contrary to Fig. 8.h and Fig. 8.j, where the move containing one or both clauses is part of an exchange. From a structural point of view, in Fig. 8.e, comes as a completion of the current parent intervention, in accordance to the rules of dialogue formation.

Regarding the Fig. 8.h and Fig. 8.j, the conjunction move is expressed by generating one of the clauses and the connective. This move comes as a reply from the second interlocutor. By using the conjunction move in Fig. 8.j, the second interlocutor reveals their curiosity.

**Succession** It shows further development of the narrative and, similarly to the conjunction, can be used as a closing of the parent move, after a sequence of exchanges, as Fig. 8.m. In Fig. 8.j and Fig. 8.h, _MSuccession_ takes part of an exchange, narrating the progression of the events from the point of view of the first interlocutor.

**Argument** In this plot are three Argument-based relations (_ARG_ in Table 1) being transposed into dialogue exchanges in Fig. 8.a, c and i. Next, we are discussing about the first relation of the plot: _ARG(PRELIM(S3), S4)_ (8.a in the dialogue). Based on its definition, we have two options for generating this part of monologue, as seen below:

- _ARG(PRELIM(S3), S4) is [Everybody knows me] Argument(TM [I am a British philanthropist]

- _ARG(S4, PRELIM(S3)) is [I am a British philanthropist] Argument(TM [Everybody knows me]_

The dialogue is constructed by adding replies corresponding to the second interlocutor. For the Argument category, there are various types of replies that can be generated. Within the dialogue generation process, we consider each case, combining it with the produced monologues. One generated solution is presented in Fig. 8.a: in this particular case, the reply consists of the actual argument by the second interlocutor, the _MARG_.

**Reformulation** A Reformulation exchange (Fig. 8.b, .d, .l in the dialogue) incorporates three text relation-based moves _M_TOPIC, M_REFORM_ and _M_COMM_, to which the replies are added. Regarding the overall effect this relation has, it may be looked at from the point of view of an interlocutor emphasising something that is happening within the plot, the discourse revolves around the topic. Out of the exchanges based on this relation, we start by discussing the exchange generated in Fig. 8.b.

This relation is equivalent to the following monologue, where the first and second interlocutors are denoted by INT1 and INT2 respectively:

_INT1: TOPIC_TM [They are not British citizens.]
_INT2: REFORM_TM [They had another child.]
_INT1: [I have a younger sister, Kamilah.]

By applying the reply strategies defined for this category, one possible outcome is to obtain the following instance:

_INT2: But they live in the UK, right?
_INT1: Yes, they live in Great Britain.

Regarding the Reformulation exchange in Fig. 8.l, the wrap-up can have different interpretations; currently, we use it to express the understanding of the information transmitted in the current exchange; since _M_TOPIC_ is there to highlight the dislocated constituent and _M_COMM_ brings in added commentary on the issue, _M_Reform_ conveys the actual information.

**Counterargument** The Counterargument is similar to Argument, but it expresses an adversarial position. For instance, in our example Fig. 8.k, Tahani admits that her life was amazing, but the second interlocutor highlights a counterexample. Another more complex example can be found in Fig. 8.i where the pattern is recursively continued to produce a stronger effect of Counterargument, i.e. by Fig. 8.f.*
Curiosity Effect (Fig. 1.c) In this example, curiosity was applied to the second move within the dialogue Exchange: it is defined by generating of the last relation in that block (part) at the beginning, as well as generating the Topicalization before that (Table 2). These constructions create curiosity by describing specific moments of change.

| g | TOPIC(S23) |
| j | SUCCESSION(CONJ(S30, S31)) |
| f | COUNTARG(PRELIM(S20), COUNTARG(PRELIM(S21, S22)) |
| h | SUCCESSION(CONJ(S26, S27)) |
| i | ARG(S29, PRELIM(S30)) |

Table 2: Sequence of Relations that are part of M_Curiosity. (Labels on the left hand side match labels in Fig. 8).

Perspectives and Future Work
In this paper, we have described a novel method and a first prototype for generating rich dialogue structures from a formalised first-person narrative description. For this proof-of-concept system, the input is currently user-edited, though we are already working on automatising and generalising some of its features, such as implementing heuristics for selecting which narrative events are good candidates for dramatic discourse manipulation, and mapping its structure to the event-based representation of the plot in use in a typical interactive storytelling system. The system will be compatible to use with a narrative generation system and NLP libraries which will provide an easier integration with interactive storytelling simulation-based applications with the ability to rely on dynamically generated dialogues between characters as another means for adapting to user interaction. A further direction to be explored is the extension of the dialogue structure building blocks, by integrating models of misunderstandings.

Acknowledgments
The work presented in this paper is funded by the Direction Générale de l’Armement, France (DGA), being carried out in cooperation with the Defense Science and Technology Laboratory, United Kingdom (DSTL).

References
Figure 8: Interlocutor 1 (INT1) – white background; Interlocutor 2 (INT2) – green background; Please refer to the main text for full details.

**Instantiated sequence of utterances from the above dialogue structure:**

**INT1:** Everybody knows me. **INT2:** As you are a famous British philanthropist. **INT1:** As for my parents, they are not British citizens. **INT2:** But they live in the UK, right? **INT1:** Yes, they live in Great Britain. **INT2:** Anyway, they had another child. **INT1:** I have a younger sister, Kamilah. **INT2:** They had another child. **INT1:** As you were not good enough for them. **INT2:** Kamilah, everybody prefers her. **INT1:** Kamilah, my sister. **INT2:** Everybody likes her. **INT1:** We both grew up in England. **INT2:** And you had the best education. **INT1:** I attended Sorbonne. **INT2:** And I went to Oxford. **INT1:** My sister, she had ordered a huge replica of her made of gold. **INT2:** At some point Kamilah’s statue squished me. **INT1:** Then I started hitting the statue. **INT2:** And it started to be unbalanced. **INT1:** The statue was so heavy. **INT2:** So it squished you. **INT1:** I have had an amazing life. **INT2:** Although the end of your life wasn’t as great. **INT1:** Regarding the end, this is how my death went. **INT1:** Basically, it happened quickly. **INT2:** It was a sudden event. **INT2:** Did you really die because of a statue? **INT1:** Then I got to the Good Place.
Creative Language Generation in a Society of Engagement and Reflection

George A. Wright and Matthew Purver
Cognitive Science Research Group
School of Electronic Engineering and Computer Science
Queen Mary University of London
{george.a.wright, m.purver}@qmul.ac.uk

Abstract
Many existing models of narrative and language generation use rigid sequences of steps which are cognitively implausible and limit creativity. Iterative models based on Sharples’ cycle of engagement and reflection improve on this by incorporating self-evaluation but still have a rigid arrangement of parts. This paper outlines how a multi-agent approach could be used to break apart the cycle into a more fluid society of engagement and reflection, whose constituent agents interact with one another to produce a text. Our approach is to work in a simple domain in order to focus on the underlying processes, and to avoid the ELIZA effect during evaluation.

Introduction
Narrative is how humans make sense of the world. A model of narrative generation is thus an important strand in the development of intelligent and creative machines. But, much AI and CC work on narrative generation focuses on efficient yet rigid generation of textual summaries and/or the generation of stories and scenarios in an interesting, literary domain. There tends to be less focus on the processes that take place in a human mind during the creation of a narrative text. This paper outlines first steps towards a model based on the interactions of micro-agents which should approximate theories of cognition such as Minsky’s (1986) Society of Mind.

The Question of Architecture
Many models of narrative and language generation use a fixed sequence of discrete steps. This is best exemplified by the data-to-text pipelines used for summarizing structured data, although neural architectures also tend to be unidirectional and run in a fixed order. The pipeline approach has been applied to many tasks, including, recently, to the description of election results (Leppänen et al. 2017). Reiter (2007) divides the data-to-text pipeline into four stages:

- **Signal Analysis** A search for patterns in the data.
- **Data Interpretation** Identification of “messages” from the patterns and relations between messages.
- **Document Planning** Selection of messages and arrangement into a rhetorical structure.
- **Microplanning and Realization** Generation of natural language text.

The stages of the pipeline explain the processes a human goes through when describing data. Indeed the work of Reiter and his colleagues is (at least in part) inspired by observations of humans (Yu et al. 2006), but the fixed, unidirectional arrangement of the processes is not realistic.

Greater realism is offered by Sharples’ (1998) cycle of engagement and reflection, partly implemented in MEXICA (Pérez y Pérez and Sharples 2001), which is divided into stages slightly analogous to those in Reiter’s pipeline:

- **Contemplate** Form ideas (≈ Signal Analysis + Data Interpretation).
- **Specify** Select and organize ideas (≈ Document Planning).
- **Generate** Produce text (≈ Microplanning and Realization).
- **Interpret** Review and interpret generated text.

The generate stage belongs to engagement, the others to reflection. The cycle restarts after interpretation, allowing for a consequent re-working of the text. This is more in tune with evidence from psychology and neuroscience that language production and comprehension are intertwined (Pickering and Garrod 2013). But large, self-encapsulated modules in fixed positions cannot fully account for this intertwining, nor for the fluidity and spontaneity we expect from what Fauconnier and Turner (2002, p321) term the “bubble chamber of the brain”. This is the case with many models, even those using sophisticated techniques for each module such as neural networks (Fan, Lewis, and Dauphin 2019) or genetic algorithms (McIntyre and Lapata 2010).

The FARG Approach More fluidity and spontaneity occurs in the models of analogy making and creativity by Hofstadter and his Fluid Analogies Research Group which consist of thousands of small agents called codelets that gradually build (and sometimes destroy) structures in a workspace (Hofstadter and FARG 1995).

One of their earlier models is Copycat, which solves analogy problems of the form “if ABC goes to ABD, what does XYZ go to?” (Mitchell 1993). Similar methods have been applied to other areas such as music understanding (Nichols 2012) and typeface design (Rehling and Hofstadter 2004).

Copycat tends to produce more sensible solutions to problems, but when faced with an unusual situation can come up with less obvious solutions (such as WYZ to the above
problem). Hofstadter compares this to the way people resist “nonstandard ways of looking at situations” unless a change in circumstances warrants it (Hofstadter and FARG 1995, p240). The usual answer to an analogy problem like the one above would be to replace the last letter with its successor in the alphabet, only in the case of XYZ that is not possible, so a more outlandish approach is taken involving a reversal.

The Copycat architecture has three main components:

The Workspace where an initial problem is perceived and structures are built by codelets to represent groupings and analogical mappings. The workspace has a temperature indicating the coherence of its structures.

The Slipnet a semantic network whose nodes spread activation and slip towards and away from one another according to the current context. Active nodes send top-down codelets to seek instances of their concept.

The Coderack where codelets are selected stochastically and according to their urgency. If the workspace has low coherence, selection is more random, and more open-minded bottom-up codelets can explore alternative paths.

In general, top-down codelets become more dominant over time as the temperature (non-monotonically) decreases and a single path to a solution is chosen. It is possible that a chosen path will result in a snag — in which case the temperature will increase, offending structures will be destroyed, and alternative pathways will be considered (Mitchell 1993).

Unlike the frameworks for language and narrative generation discussed above, FARGitecture does not involve a central authority directing the model through stages in a sequence: control is distributed between codelets and slipnet nodes. When more bottom-up codelets are running, the system is in a relative state of reflection (contemplating new structures and reviewing existing ones), while when more top-down codelets are running, the system is in a relative state of engagement (pursuing a particular path towards a solution). FARGitecture therefore enables a fuzzy alternation between engagement and reflection.

Copycat’s lack of central control, tendency to vary its behaviour due to stochasticity, and ability to pursue stranger solutions when circumstances allow make its architecture more cognitively plausible than other more rigid models.

The Question of Domain

This paper outlines how ideas developed by Hofstadter and FARG (1995) could be applied to narrative generation. Their approach is to work in micro-domains so that evaluation must focus on the decisions a program makes while exploring its search space, not on any meaning inherent to the space. This is a different approach from most work in creative language generation which tends to cite Meehan (1976) as the earliest work in the field while overlooking the more modest (yet more impressive) work of Davey (1974). Whereas Meehan’s TALE-SPIN generates stories about animals living in a forest, Davey’s PROTEUS narrates games of tic-tac-toe. PROTEUS’ subject matter is boring but its use of features such as co-reference and conjunctions produces highly readable pieces of text. TALE-SPIN, on the other hand, outputs stories as lists of self-contained pseudo-English sentences which are easy to understand but aesthetically displeasing. Work on creative language generation tends to deal in overtly literary domains. But, all language is creative: even a tic-tac-toe commentator has to make decisions about how to structure a text; how terse or detailed to be; and what words to use where.

At this early stage in the path towards creative machines, research should avoid complex, literary domains which give the impression of creativity where there is none, and first see how decisions can be made in a simpler domain of discourse. This will prevent evaluators from succumbing to the ELIZA effect — jumping to the conclusion that a machine has achieved human levels of intelligence when it really only relies on a few simple tricks. Veale (2017) shows that, when using the same method to build plot skeletons, giving characters the names of celebrities results in higher ratings for dimensions including imagination and drama than when using generic animal characters. Readers cannot help but find meaning in a text which the artificial author is oblivious to.

Following FARG and Davey, this paper outlines a proposed architecture for narrative generation intended for testing on mini-domains such as weather and board games.

Describing a day’s weather forecast involves recognizing entities such as storms and patches of warm or cold weather; tracking their movements and changes; and weaving together these threads into one linear piece of text. Certain aspects of narrative are lacking from this domain: for example, there is no need to account for characters or their motivations. But describing the weather does require many mechanisms fundamental to narration: formulating a narrative of the weather requires the ability to select interesting pieces of information; discard other pieces; find appropriate names for the entities that have been recognized; and to find a good structure for the text. There are many non-trivial issues to tackle — even in this simple domain.

Board game narration is a domain that could provide some of the other ingredients of narrative: there are characters with goals and plans (the players), and there is space for imagined counterfactuals. In some ways board games are simpler than the weather: entities in checkers and chess are discrete whereas weather patterns have fuzzy boundaries. Board games also have a clearer beginning and end.

Ultimately, an architecture that could handle both of these domains would be a good candidate for a general model of humans’ storytelling capacity. This paper focuses, for the most part, on the domain of weather.

A Society of Engagement and Reflection

In this (yet unimplemented) architecture everything is done by codelets, including: data interpretation; arrangement of the text; language realization; evaluation of structures; and destruction of those that are no longer wanted. These tasks correspond to the modules in pipeline and cyclic architectures discussed above, but while most models perform these functions in a strict order, in this society model the tasks are broken down into small units of work which can be carried out whenever appropriate. A codelet runs not according to its position in a line-up, but due to competing data-driven
bottom-up pressures and conceptual and aesthetic top-down pressures.
Each codelet can be classed as either bottom-up or top-down. Bottom-up codelets are more open-minded, looking for anything of interest, whereas top-down codelets are more single-minded, looking for instances of a specific concept.

Data Labeling and Grouping Codelets Bottom-up data interpreting codelets access raw data in the workspace and determine the best concept with which to label it. For example, in the weather domain, a location with a temperature of 25°C may be labeled HOT. This leads to the HOT semantic network node receiving a boost in activation. Once fully activated, this node sends out top-down codelets to look for other locations that can be labeled as HOT. After a while, many of the same labels begin to appear in one region of the map and grouping codelets, recognizing the similarity, divide the map into regions corresponding to weather type.

These codelets perform a similar role to a convolutional kernel in a neural network, indeed they could each be implemented as a neural or other machine learning classifier. The benefit of using individual codelets which are run according to the urgency determined by activations in a semantic network, instead of having fixed layers in a neural network, is that they are not necessarily run unless the combination of context and top-down desires deems it necessary. For example, having recognized a pattern of interest in the north of a map, the NORTH node in the semantic network may spread activation to the SOUTH node to encourage a search for a pattern which summarizes the south. This architecture of interacting codelets allows for higher-level relational processing to be followed by a reversion to lower-level raw-data processing similar to how Yu et al (2006) found experts switch between more coarse and more detailed views when analyzing data to get “details-on-demand”. Feed-forward neural architectures and traditional pipeline architectures, on the other hand, rely on all of the data interpretation that could possibly be relevant having been done at an early stage.

Language Generation Codelets Several codelets perform the task of microplanning and realization.

Phrase codelets recognize a structure that can be transformed into a phrase. E.g. rainy → It will be rainy.

Connective codelets recognize two phrases which can be joined. E.g. It will be rainy. It will be cold. → It will be rainy and it will be cold.

Deletion codelets remove unnecessary parts of a phrase once it has been connected. E.g. It will be rainy and it will be cold. → It will be cold.

Ordering codelets order two or more phrases or sentences, such as in a general-to-specific order or along a dimension of a conceptual space. E.g. It will be warm in the midlands. It will be hot in the south. It will be cold in the north → It will be cold in the north. It will be warm in the midlands. It will be hot in the south.

Phrase codelets essentially apply templates. But, the aim is to limit the size of templates and allow for them to be combined, re-ordered and re-structured in order to limit repetitiveness. This is similar to the approach taken by Leppänen et al (2017), but this architecture should allow for more diverse realizations. For example, there may be different ways to order phrases according to the most salient concepts in the context; and there may be different ways to connect phrases according to how ordinary their co-occurrence is: hot but rainy makes sense; cold but rainy does not (at least from a British perspective). The exact realization that the architecture chooses will in part depend on its stochasticity and it will not be expected to re-produce the same text if run again.

Other codelets are also required, such as those that arrange rhetorical structure and those that pick which information to include in the text.

A Hypothetical Example

Figure 1 is an example of a map of the weather at a point in time for the model to describe (more realistically, it should handle a sequence of maps in order to qualify as narrative). This map has four channels: weather type, wind (direction and speed in kph), temperature (in centigrade) and percentage probability of precipitation. Below is an example of a textual forecast it might generate.

It will be cloudy in the north with a high chance of rain and furthermore snow in the very north. There will be dry weather in the rest of the country but there may be pockets of rain in the south. It will be sunny in western and central areas but temperatures will be mild while it will be cloudy but warm in the southeast.

![Figure 1: A four-channel map of weather in Britain with groups and relations. 1-12: Regions of similar weather; 13: AND relations; 14: BUT relations; 15: A FURTHERMORE relation; 16: A second-order AND relation. (Data from the Met Office).](image-url)
At the start of the program’s run bottom-up codelets search for the types of weather present on the map. Codelets also group them into regions. The ellipses in figure 1 indicate approximate regions that might be recognized. Codelets then find relations between regions. Certain regions are recognized as being to some extent the same, for example regions 2 and 8 in the north of the country. The north’s cloudy weather and high chance of rain are ordinarily co-occurring types of weather thus are connected with AND. Meanwhile the south’s cloudiness and warmth are less typically so are connected by BUT. When a sub-region has a more extreme kind of weather than its parent region, for example the snow in a small part of the north, a FURTHERMORE relation is used. When a temporal sequence of events is being described, yet more relations can be recognized, such as THEN and THEREFORE. Higher-order relations are also possible: 16 shows an AND connecting two parallel BUTs.

Codelets use weather, location, and relation labels to begin forming phrases. Certain labels depend only on local concepts such as “the north”, while others such as “the rest of the country” are context-sensitive. Arrangement of the text also depends on linguistic context. For example, the sentence describing the rest of the country must come after the sentence describing the north in order for the rest to make sense. The sentence comparing the western and central areas and the southeast ought to come last since it is an elaboration of the sentence describing the rest of the country. Codelets must recognize the importance of context and discourse relations as they arrange the final text.

Open Questions

Many questions need to be answered in order to get this architecture working: what conceptual knowledge will the model require? Can the model be applied to board game narration and beyond? How much of the workspace context must each codelet be aware of? How will the model handle complex situations where concepts have varying relevance in different places?

This last issue, French (1995) describes as the “problem of single nodes with multiple activations”. It was a major problem in his (FARGitecture based) model of analogy making between objects on a dinner table, and required a hierarchy of different contexts corresponding to different patterns of activation in the semantic network. It is likely to be an even larger problem in narrative formation, which can involve summarizing even more situations than when making a single analogy.

Conclusion

There remain issues to be resolved in applying this style of architecture to narrative generation, but its potential for flexibility makes it an attractive line of research. Work so far has centred around the mundane domain of weather so that focus can be placed on the most fundamental issues involved in narrative and language. Future work should move into richer domains such as board game narration in order to better test the generality of the approach.

Acknowledgments

This research has been supported by EPSRC grant EP/R513106/1 and by the European Union’s Horizon 2020 research and innovation programme under grant agreement No 825153, project EMBEDDIA (Cross-Lingual Embeddings for Less-Represented Languages in European News Media). The results of this publication reflect only the authors’ views and the Commission is not responsible for any use that may be made of the information it contains.

References

Meta-level Evaluation and Transformational Creativity; An analysis of MEXICA

Juan Alvarado\textsuperscript{1} and Geraint A. Wiggins\textsuperscript{2,1}

\textsuperscript{1} School of Electronic Engineering and Computer Science, Queen Mary University of London, United Kingdom
\textsuperscript{2} Department of Computer Science, Vrije Universiteit Brussel, Belgium
\{j.alvaradolopez, geraint.wiggins\}@qmul.ac.uk, Geraint.Wiggins@vub.be

Abstract

We present a proposal for meta-level evaluation in creative systems. In this paper, the Mexica creative system is the subject of our analysis. To achieve meta-level evaluation, we have constructed a program that interacts with Mexica’s inputs and its outputs. The meta-level system evaluates the results and makes decisions based on them. For decision-making it includes heuristics that are defined in terms of Mexica’s results; these are rules particular to the object-level system. We also include other general metrics that apply in any other creative system. For example, Ritchie’s criteria for evaluating creative systems are included in this proposal.

Introduction

Creative systems have been developed for various purposes and fields. We can find them for the generation of stories, melodies, paintings, poems, etc. There are also creative systems for non-artistic domains because creative behaviours are important here too.

There are many things to analyse in creative systems, for example, the outputs they produce, the rules, the input examples and the evaluation methods they use, etc. An important aspect of the analysis is the evaluation of the work of these systems. One way to approach the evaluation of a system is in terms of the output it produces (Ritchie 2007), we can call this an Output Evaluation. Another way to evaluate it is to analyse the process that the system follows (Colton, Pease, and Charnley 2011; Jordanous 2012), we can call this a Process Evaluation. We could also consider a combination of both types of evaluation where we analyse both the output and the process, we can call it a Process-Output Evaluation. Humans can evaluate the creativity of systems, and creative systems can evaluate their performance themselves (Human Evaluation, System Evaluation). There also could be a meta-evaluation; an evaluation of their evaluation (Jordanous 2011).

There is another way to evaluate creative systems, which is to go to the meta-level. In the meta-level there are rules, but they do not need to be the same as at the object level. As an example, take this paper. It was generated by an actor (one/many authors, a system, etc.). The paper is the output. The author has rules to generate a conceptual space, find novel concepts and evaluate them. This is the object level.

In the meta-level, a reader looking at this paper might have a distinct set of rules and different knowledge. Therefore, the evaluation of the paper might be different as this meta-level evaluation might not share the evaluation rules in the object level. Following the terms suggested before, assuming the reader is a human being, this would be a Human Meta-Level Output Evaluation.

In this paper, we take the creative system Mexica as an example and explore its meta-level output evaluation. To achieve this, we have built a new system that interacts with Mexica. The meta-level system interacts with the graphical interface of Mexica and the outputs it generates and changes the parameters Mexica uses to produce an output. Therefore, we are talking about a System Meta-Level Output Evaluation.

To achieve meta-level evaluation, we have to consider the distinct parts of Mexica. For this we have used the ideas on creative systems expressed by Boden (1990) and the Creative Systems Framework (CSF) proposed by Wiggins (2006). With these tools, we can formalise the elements of Mexica in a model that identifies its components and how they interact.

We also use the criteria proposed by Ritchie (2007) for evaluating creative systems. These criteria can give us information about the typicality and quality of stories generated by Mexica. Using this evaluation along with the information from the files Mexica generates, we can know, for example, the distribution of stories in the groups proposed by these criteria, the characteristics these stories have, the result of the self-evaluation for each story and the set of parameters used to generate them. This information allows us to build a meta-level system that admits transformational creativity.

This paper is about how to do reflection in a creative system as suggested by Buchanan (2001), structured as transformational creativity in Boden (1990) terms. One of our reviewers summed up the intent of this paper, extremely well: “The result is a meditation on computational creativity, transformational creativity and conceptual spaces that is relevant to the topic.”

Background

We have brought together the following ideas in this work. These are important because we use them to distinguish the distinct parts of the creative systems and how they interact.
This way, we are in a better position to perform meta-level evaluation and later modification in Mexica and thus achieve transformational creativity.

**Conceptual Spaces and Transformational Creativity**

Boden (1990) has expressed various ideas about creative systems. One of the key components of her ideas is conceptual spaces. They contain the creative ideas that systems can find. Conceptual spaces, as expressed by Boden, are spaces delimited by accepted rules in a social group. They define the space of solutions of a creative system.

Boden (1990) also points out that there are distinct types of creativity in systems. One of the important types of creativity that she points out is transformational creativity. For Boden, this type of creativity is characterised by changing the rules that define the conceptual space (Boden 1990). This means that the accepted rules in a social group have changed and therefore the concepts that can be found in a conceptual space have also changed. This could mean the existence of more, fewer, better concepts. In any case, it means a change in the conceptual space and a change in the concepts or ideas available to a particular system.

Wiggins (2006) adds to the proposal of Boden (1990) that we can have transformational creativity by changing not only the conceptual space but also the rules a system uses to traverse it. This way we are also making different concepts or ideas available. They could be more, fewer, better, etc. but the important thing is that there has been a transformation.

**Mexica**

Mexica (Pérez y Pérez 1999) is a creative system that generates short stories. Mexica does not have pre-defined goals. It uses an Engagement and Reflection cycle (Pérez y Pérez and Sharples 2001; Sharples 1996). Content generation is guided by a set of constraints and previous stories available.

Mexica includes different rules for Engagement and Reflection. Alvarado and Wiggins (2018a) point out that in the original Sharples (1996) account each phase produces different results because conceptual spaces are not the same. Similarly, Alvarado and Wiggins (2018b) point out that Mexica produces a story through the interaction of both phases in which conceptual spaces are not the same either. In one phase, the system generates a type of content which, passing to another phase, is restricted and changed.

In Reflection, the system evaluates the story in progress. The evaluation takes into account the parameters provided by the user. This can lead to the system changing parameters to improve the story. For example, if the story is getting boring or if the evaluation determines that the story in progress is very similar to one/some previous stories in the database.

The system delivers as a final output a story and several files that show part of the process that Mexica followed to generate the story. In these files, the system shows the parameters it has used, the actions it has added to the story and the results of the self-evaluations.

**Creative Systems Framework**

Wiggins (2006) has proposed a framework for creative systems. In this framework, he proposes a formalisation of creative systems in such a way that we can identify different aspects in a creative system and the rules that operate on them. This model takes into account ideas of Boden (1990) for conceptual spaces and formalises them. Wiggins (2006) highlights the rules systems use and the concepts that can be found using them in conceptual spaces.

In this framework, Wiggins (2006) points out that there are rules that a system can use to define its conceptual space. He takes very much into account the process that the system follows for the generation of artefacts. He considers a set of rules that the system uses to explore the conceptual space. He also considers the use of a set of rules by which the system can determine the quality of the objects produced. Different systems apply different rules, and their number is variable.

Wiggins (2006) suggests an interpretation function that uses the mentioned rule sets to find creative objects. We can change rule sets to change the behaviour of a system. This results in many interesting behaviours. For example, rules that define the conceptual space can change and they can generate new conceptual spaces. The rules that serve to traverse the conceptual space can change, giving rise to having access to new concepts which were previously inaccessible. In principle, it would be possible to change the evaluation rules, allowing previously rejected ideas to be accepted.

**Ritchie’s Criteria**

Ritchie (2007) has proposed some criteria for evaluating creative systems based on the quality of the output that they produce. These criteria are useful for finding the proportion of examples that fall into different important categories in creative systems analysis. For example, the percentage of new artefacts to the examples known to the system, or the percentage of artefacts that are high-quality and untypical.

In this paper, we also suggest the use of this set of criteria to feed the meta-level system with information on the evaluation of the products generated by Mexica. This way, the meta-level system can make decisions about the parameters that can be varied to change the behaviour (and outputs) of Mexica.

Alvarado and Wiggins (2019) propose a model that integrates the Engagement and Reflection account, with the Creative Systems Framework and Ritchie’s criteria. Here the similarities between the CSF and Ritchie’s criteria are highlighted. A particular example of the coincidences between the two proposals is the importance given to the generation of untypical but good quality examples, which could be seen as an achievement for creative systems. This is so because untypical examples generally correspond to solutions that are not found in the conceptual space and evaluation processes, taking into account only this, could show they are poor quality. Finding untypical and good quality artefacts implies that the evaluation rules have been able to find good quality artefacts despite not being in the conceptual space. Wiggins (2006) calls this aberration to highlight the negative
connotation with which artistic examples of good quality but not following pre-established rules have been received in the past.

Alvarado and Wiggins (2020) show the application of Ritchie’s criteria to the output generated by Mexica. They highlight here the outputs’ trends for Mexica. This can be useful when evaluating an instance of Mexica or deciding how to find a new instance.

**Implementation**

For the implementation of this project, a program that can interact with the Mexica’s graphical interface and its output has been built. The purpose of this program is to change the input of Mexica based on a meta-level evaluation.

**Object level**

At the object level, we have Mexica and its components. These include inputs, outputs, and rules that it uses. As inputs, we have the initial action that the system uses as a starting point, the previous stories and a set of parameters that are defined in its user’s interface. The outputs include the log files, the generated story and the set of all stories generated in previous runs of Mexica. The log files contain information on the process that the system has followed to generate the stories.

Alvarado and Wiggins (2018b) have analysed Mexica and identified distinct groups of rules for Engagement and Reflection. They argue that these different rules produce different results in each stage.

Before Mexica generates a story, it analyses all previous stories. It takes each action in each story and computes the context. Mexica creates structures with the generated contexts. It groups all actions in previous stories that share the same context into the same context structure.

When Mexica generates a story, it receives an initial action. Mexica computes the context for the initial action. Then, using this context, it looks for a similar one among the context structures. Mexica retrieves all the associated actions from those similar contexts. Those actions will be candidates to continue the story. It generates something similar to a tree with a variable number of branches to continue the story. It analyses each branch with other conditions that filter the candidates. After the filtering, Mexica selects one of the remaining candidates to continue the story in progress. Mexica computes the context of the story in progress with each new action added and repeats the process in each execution of Engagement.

Context similarity checking requires some parameters the user can set in the graphical user interface. The number of matches between the context structures (and potential candidates) depends on the value of these parameters. Candidates filtering also requires parameters the user can set in the graphical interface.

The lack of options to continue the story (i.e. there are no matches with context structures) can cause an impasse. The reflection stage has mechanisms to solve this problem, but if it cannot do so, the system gives up and the story is abandoned.

**Interface**

To interact with the object level, the meta-level system requires an interface we have built. This interface is a program capable of interacting with Mexica’s graphical interface, changing parameters and starting its execution. This interface has access to Mexica’s output files. It reads from the object level the previous stories, the log files, the generated story, the set of all generated stories and the set of parameters defined by the user. This interface communicates directly with the meta-level system. The output of this interface (which comes from the meta-level system), includes parameters and the initial action to adjust the future behaviour of Mexica.

**Meta level**

Wiggins (2006) argues that we can view transformational creativity as exploratory creativity at the meta-level. This way we can use the same components he uses in the analysis of creative systems in the meta-level.

In the meta-level system, some rules define the conceptual space. Here, the conceptual space corresponds to the space of all possible instances of Mexica. Mexica’s user inputs constraint this space. The user in this case is the meta-level system. It only changes the parameters, not the previous knowledge.

A set of rules allows the meta-level system to traverse its conceptual space. They do so by varying Mexica’s parameters. With this, they generate a new instance of Mexica. These rules allow decision-making with the information provided by the meta-level evaluation.

There are meta-level evaluation rules that can be used with instances of Mexica found in the meta-level conceptual space. The evaluation rules in the meta-level include heuristics that take into account the information in Mexica’s log files (e.g. the result of the self-evaluations, or the number of actions retrieved in the first execution of Engagement). For example, they take into account the probability of success of a story in progress when there is an impasse in the first execution of Engagement, which is low.

Following this example, when the meta-level system detects an impasse in the first execution of Engagement, it raises a flag because of the meta-level evaluation. Then using the traversing rules, this evaluation and the current set of parameters, the meta-level system makes a decision. As a result, the meta-level system establishes a new value for the similarity of contexts parameter, so Mexica can get more options. With this, using Mexica’s graphical interface, the interface sets the new set of parameters and a new initial action and runs Mexica. This way, in this example, Mexica can create a story, avoiding the impasse because now it has more actions (or different) to choose from. This modification considers the self-evaluation of Mexica and the meta-level evaluation to produce a story that gets a better overall evaluation. The purpose of the meta-level system is to make the evaluation at the object level be satisfied more and more often.

Alvarado and Wiggins (2020) show the result of applying criteria of Ritchie (2007). They provide performance-related...
information on past Mexica runs. The results show that there are not untypical and good quality examples produced by Mexica. We can include this result in the meta-level evaluation and traversing strategy to make a more informed decision on how to change parameters to get better stories, but more than that, explore the possibility of finding untypical and good quality concepts.

Discussion
We have shown one way that we could meta-level evaluate creative systems. Here, we have used Mexica, a creative system that produces stories. We can think of many examples of creative systems. They follow different processes and approaches to be developed and use different rules. What we present here is an idea and a way of doing meta-level evaluation in a particular example.

Pérez y Pérez (1999) points out that changing Mexica’s parameters offers great flexibility to experiment with the creative process. What the meta-level system does is precisely this; it interacts with Mexica, examines the output, evaluates at the meta-level, decides and establishes new parameters for Mexica, and it runs Mexica again. It interacts as a user so it cannot change the inner workings of Mexica.

In this paper, we propose to use the criteria of Ritchie (2007) to determine Mexica’s performance. The criteria form a good tool that gives information regarding the outputs that Mexica generates. While their first intention is to be used by humans to evaluate the outputs of the systems, Ritchie (2012) points out that it is possible to use them as internal components of the system and not external judgements. This means that we can apply these criteria in distinct ways: Internal: How does the system work, on its own terms? External: How does the system work, in terms of independent measures such as human judgements? Considering the internal way, we can do what we report in this paper. It should be possible to use these criteria as part of a system that evaluates another system. There is still the external part in which we should include human intervention to evaluate the meta-level system and its meta-level evaluation.

Ritchie’s criteria do not have a particular field of application, which makes them a generic part of this meta-level system. Perhaps it is necessary to include other general approaches/methodologies for the evaluation (e.g. Colton, Pease, and Charnley 2011; Jordanous 2012).

There are other elements of this meta-level evaluation that are not general. For example, they largely depend on the results that Mexica delivers and on the results of its self-evaluations. The same happens with other creative systems. Further refinement is necessary to incorporate general ideas in creative (meta-level) (meta) evaluation to improve this proposal.

Acknowledgements
We would like to thank the reviewers for their comments which helped improve this paper. Thanks to the National Council of Science and Technology (CONACYT) in México for its sponsorship. The research was also supported by funding from the Flemish Government under the “Onderzoeksprogramma Artificiële (AI) Vlaanderen” programme.

References


Discovering Textual Structures:
Generative Grammar Induction using Template Trees

Thomas Winters and Luc De Raedt
Computer Science Department, Leuven AI
KU Leuven, Belgium
{firstname}.{lastname}@kuleuven.be

Abstract

Natural language generation provides designers with methods for automatically generating text, e.g. for creating summaries, chatbots and game content. In practice, text generators are often either learned and hard to interpret, or created by hand using techniques such as grammars and templates. In this paper, we introduce a novel grammar induction algorithm for learning interpretable grammars for generative purposes, called GITTA. We also introduce the novel notion of template trees to discover latent templates in corpora to derive these generative grammars. By using existing human-created grammars, we found that the algorithm can reasonably approximate these grammars using only a few examples. These results indicate that GITTA could be used to automatically learn interpretable and easily modifiable grammars, and thus provide a stepping stone for human-machine co-creation of generative models.

Introduction

Text generation is a prominent tool within computational creativity, due to many creative fields using text as its primary medium, e.g. poetry and humor. As such, many computational creative systems have applied a large variety of text generation methods. Methods like templates and grammars generally grant the model designer relatively more control over the output, but tend to be labor-intensive to create. Neural text generation approaches (e.g. RNNs, transformer models) on the other hand can create impressive language generators, but at the cost of predictability, interpretability and ease of regulating its outputs in a directed way. In this paper, we explore a new technique for learning grammars for generative purposes by discovering latent templates such that the grammar is easily interpretable and modifiable by designers of generative textual models for creative purposes.

Background

Templates

Using templates is a popular approach for generating text. In this context, a template is a piece of text with several slots that are later filled in using a particular data source. While they have a lot of obvious merits for e.g. chatbots and front-end web development, templates are also popular in computational creative applications. For these creative purposes, templates are often paired with schemas for providing sensible content to the templates that internally encode how the template slot values relate to each other (Bird and Ritchie 1994; Winters, Nys, and De Schreye 2018). For example, the expansion of non-terminal S in Figure 1 could be seen as a template, where a schema (not in the figure) would create sensible pairings for T and F. There have been several efforts into automatically learning such templates and schemas from single examples by analysing linguistic relationships and properties (Hong and Ong 2009; Winters 2019a).

Generative Grammars

Grammars are another popular way of generating text. A context-free grammar (CFG) is a four-element tuple $$(V, \Sigma, R, S)$$, where $$V$$ is a finite set of non-terminals, $$\Sigma$$ a finite set of terminals, $$R$$ a set of production rules that map elements of $$V$$ to $$(V \cup \Sigma)^*$$ and $$S$$ the start symbol. While generative grammars were initially mainly used for generating according to a text need, they are also used for creative purposes. Tracery is a popular language among casual creators for designing generative grammars. Such grammars usually extend CFGs, e.g. adding stored assignments and rule weights (Compton, Kybartas, and Mateas 2015; Winters 2019b). A prominent design pattern in these grammars is specifying production rules that map non-terminals either to templates, or to a list of possible values for a particular template slot (Figure 1). The grammar then fills the templates with randomly generated slot value combinations.

Figure 1: An example grammar capable of generating 12 different sentences specifying (odd) dish toppings.

Learning Grammars

There are many different algorithms for inducing CFGs, usually designed for a particular class of grammar. The most popular type of grammar induction induces part-of-speech tag structures from treebanks or plain text. Another popular
type of grammar induction is discovering repetitive structures to help encode the input text efficiently, inducing grammars where each non-terminal has only a single production rule (Nevill-Manning and Witten 1997). The grammars induced by these algorithms are however shaped differently than typical generative grammars with the template-like production rules. The latter generally avoids recursive production rules, as generating texts of unbounded lengths is usually undesirable for the creative goal. Non-recursive grammars are thus tools for compactly specifying a finite space or interesting texts. In this paper, we introduce an algorithm that can learn such non-recursive context-free grammars using a template-focused approach, which can thus easily be interpreted and adapted by generative grammar creators.

**Template Trees**

We create and define the notion of template trees as an intermediary step for inducing a generative grammar, and propose an algorithm for learning template trees from input text.

**Template Tree Definition**

A template tree is a connected acyclic directed graph where each node represents a template that is more general than the template of all its child nodes, thus defining a partial ordering. The leaves of a template tree are templates without slots, i.e., the input sentences used to learn this tree. A template slot maps to zero or more other template elements (i.e., slots and/or word tokens). A simple template tree can be seen in Figure 2.

![Figure 2: A template tree example](image)

**Learning Algorithm**

1. **Learning Algorithm**

   **Algorithm 1** Calculating template tree from input texts

   **Require:** input texts $D$
   
   **Ensure:** $A = \{t\}$ where leaves$_t = D$

   $Q \leftarrow \{(d_i, d_j) | d_i, d_j \in D\}$
   
   $A \leftarrow D$

   while $\#A > 1$
   
   $M \leftarrow \arg\min_{(t_i, t_j) \in Q}(d(t_i, t_j)) \cap \{(a, b) | a, b \in A\}$
   
   $N \leftarrow \{\}$

   for all $(t_i, t_j) \in M$
   
   $m \leftarrow merge(t_i, t_j)$
   
   $N \leftarrow N \cup \{m\}$
   
   $A \leftarrow A \setminus \{t_i, t_j\}$

   end for

   for all $n \in N$
   
   $Q \leftarrow Q \cup \{(n, a) | n \in N, a \in A\}$
   
   $A \leftarrow A \cup \{n\}$

   end for

   end while

**GITTA: Template Tree to Grammar**

We introduce a new grammar induction algorithm named **GITTA (Grammar Induction using a Template Tree Approach)**. **GITTA** aims to induce a non-recursive CFG, thus compactly representing a finite number of similar finite strings. While any finite language of size $n$ can trivially be represented by a simple grammar with $n$ production rules, having fewer production rules implies that patterns have been induced. This allows the grammar to potentially generate unseen examples from the language, and also be more easily modifiable. **GITTA** converts the template tree into a grammar by assuming independence between slot values, and simplifying the template tree. The resulting slot values and root template then specify the grammar.

**Pruning Template Tree**

**GITTA** first prunes redundant children of template tree nodes. A child is redundant if all its descendant leaves are reachable through the other children. For each level, nodes
are checked in ascending order of number of descendants, pruning nodes with less general templates first.

Merging Slots
To convert the template tree into a grammar, GITTA assumes that all possible slot values are independent from all other slot values of the template. For every slot, all possible slot values are extracted from the templates of the children of the nodes having a template with this slot.

After finding all slot values \( U_i \) for every slot \( s_i \), the algorithm merges similar slots if \( \#(U_i \cap U_j) \geq r \), where \( r \in [0, 1] \) is determined by the user. Lower values of \( r \) thus require slots to have less overlap in slot values in order to be merged. GITTA also removes \( u_j \in U_i \) if there is a slot \( s_k \) such that \( s_k \in U_i \) and \( u_j \in U_k \). If \( s_i \in U_i \), the slot will also be removed from the slot values. If \( U_i = \{s_k\} \), then \( s_i \) is replaced with \( s_k \). For example, for the tree in Figure 2, the algorithm would discover that \( D \) has the same slot values as \( B \), and thus should be replaced by \( B \). This process continues until there is an iteration without any update.

Collapsing Template Tree
Using the merged slots, several simplifications are made to the template tree. First, the replacement mapping reduces the number of different slots of the template tree. Second, knowing the slot values for a slot helps reduce the number of the number of direct children of a node in the tree. For a node \( p \) with template \( t_p \), and a child \( c \) with template \( t_c \) that contains a slot of \( t_p \) and for which the template \( t_c \) can be obtained by filling in other slots with known slot values into \( t_p \), then this child node \( c \) is redundant and can be pruned. All children of \( c \) are then added as direct children of \( p \). For example, for Figure 2, if the root template would be \( \langle C \rangle \langle B \rangle \langle B \rangle \), the four middle nodes would collapse into their parent, leaving only \( \langle C \rangle \langle B \rangle \) as parent of the four leaf nodes. After collapsing the template tree using knowledge of the slot values, the template of each node is recalculated, which leads to the aforementioned new root node template of Figure 2. This process of simplification of the template tree and recalculation of the templates keeps repeating until the template tree is unchanged after an iteration. The resulting grammar is derived by mapping from start symbol \( S \) to the root template of the template tree, and using slot values mappings as production rules.

Experiment: Reverse-engineering Grammars
To measure the performance of the algorithm, we test how well it can induce grammars that generate significantly more elements of the original language than shown as example to the algorithm, with usually relatively few elements not in the original language. However, GITTA also sometimes uses relatively more rules \( R_I \) to generate relatively less generations \( L_I \), most notably in grammars 1 and 2. This indicates that many rules are likely redundant or should be decomposed into simpler rules to allow for more generations. For grammar 6, generalisation is not possible due to the origin template having one slot, and this slot mapping to different word lists, which also explains why it has more production rules than generations.

For grammars 4 and 5, GITTA tends to induce grammars with relatively large numbers of generations that are not in \( L_G \). This is usually due to overgeneralisation. For example, a grammar \( G \) that has the production rule \( S \rightarrow \langle Hello \rangle \langle World \rangle \mid \langle Hello \rangle \langle there \rangle , \langle Name \rangle \rangle \), might lead to GITTA creating a more general rule \( S \rightarrow \langle Hello \rangle \langle There \rangle \langle Thing \rangle \), with “There” \( \rightarrow \langle there \rangle \mid \langle e \rangle \) and “Thing” mapping to all values of “Name” and “World”. For grammar 5 in particular, the origin template has four consecutive non-terminals separated from two other non-terminals by only one terminal, all mapping to varying number of terminals. This property makes it unclear for GITTA where slots start and end, thus leading to overly specific production rules being added instead of finding clear slot values.

Discussion & Future Work
GITTA could be employed in a collaborative generative grammar building tool, where a designer and the algorithm create a generative grammar together. In this scenario, the designer could first illustrate several examples or use an existing corpus specifying what the grammar should generate, for which the algorithm will propose a suitable grammar by discovering latent templates, thus creating an initial grammar prototype. The designer can then add, remove and modify production rules to further suit their needs, thus allowing more meaningful interactions than black-box generative text generators generally allow. This direct control could be used e.g. for limiting the possibilities of generating offensive or

\[ \text{https://botwiki.org/} \]
\[ \text{https://cheapbotsdonequick.com/} \]
unwanted content, which is an important aspect for many text generation domains such as game development.

One limitation compared to other grammar induction algorithms is that it cannot induce recursive grammars. As such, production rules like $S \rightarrow SS \mid (S)$ ($\epsilon$ (= the bracket language) are not able to be learned by our system. However, since recursion is generally an unwanted property of generative grammars due to making grammars able to generate unbounded texts, our proposed algorithm thus prevents language model overgeneralization caused by recursion.

GITTA creates a basis for learning more complex, interpretable generative models. It could be trivially extended by learning probabilities of rules as a post-processing step using the input sentences. Another interesting extension is learning constraints that hold between expansions of non-terminals, and thus create complex generative schemas.

We mainly see the use for this algorithm in automatically mimicking patterns or extending data sets that have some sort (possibly latent) template in their texts, such as forum topic titles or writing and comedy prompts. Template trees in itself could also be used for discovering frequently occurring templates in a corpus, and provide similar functionality as clustering algorithms. The code of GITTA is available on https://github.com/twinters/gitta.

<table>
<thead>
<tr>
<th>Grammar $G$</th>
<th>$I$ from 25 examples</th>
<th>$I$ from 50 examples</th>
<th>$I$ from 100 examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Name</td>
<td>#LG</td>
<td>#RG</td>
</tr>
<tr>
<td>1</td>
<td>botdoesnot</td>
<td>380292</td>
<td>363</td>
</tr>
<tr>
<td>2</td>
<td>BotSpill</td>
<td>43452</td>
<td>249</td>
</tr>
<tr>
<td>3</td>
<td>coldteabot</td>
<td>448</td>
<td>24</td>
</tr>
<tr>
<td>4</td>
<td>hometapingkills</td>
<td>4080</td>
<td>138</td>
</tr>
<tr>
<td>5</td>
<td>InstallingJava</td>
<td>390096</td>
<td>95</td>
</tr>
<tr>
<td>6</td>
<td>pumpkinspiceit</td>
<td>6781</td>
<td>6885</td>
</tr>
<tr>
<td>7</td>
<td>SkoolDetention</td>
<td>224</td>
<td>35</td>
</tr>
<tr>
<td>8</td>
<td>soundesignquery</td>
<td>15360</td>
<td>168</td>
</tr>
<tr>
<td>9</td>
<td>whatkilledme</td>
<td>4192</td>
<td>132</td>
</tr>
<tr>
<td>10</td>
<td>Whinge_Bot</td>
<td>450805</td>
<td>870</td>
</tr>
</tbody>
</table>

Table 1: Grammar induction results given a specific number of random generations of $G$, measuring median number of generations of the induced grammar $I$ that are in and not in the target language, as well as their median sizes, over five runs.

References


Computational Humor: Automated Pun Generation

Bradley Tyler, Katherine Wilsdon, Paul Bodily
Computer Science Department
Idaho State University
Pocatello, ID 83209 USA
{tylebrad, wilskat7, bodipaul}@isu.edu

Abstract

Humor is incorporated into our daily interactions, but conceiving jokes ideas can be inherently difficult to produce spontaneously. The PAUL BOT system, “pun are usually lame but occasionally terrific”, includes features of the JAPE system by incorporating aspects of the SAD generator, templates, and sentence forms to automatically generate a pun (Ritchie 2003). A two-word database is composed of adjective-noun phrases that contain a homophone. The punchline to the pun is replacing a word in the two-word phrase with a homophone. A synonym is derived from the homophone, and a hypernym is chosen from the non-homophone word in the original phrase. The synonym and the hypernym are incorporated into a predetermined sentence structure to construct the question. Surveys were conducted to evaluate our artefacts produced by PAUL BOT in order to identify future improvements of the system. The creativity in the system is attributed to the novelty of the unique artefacts, the level of surprise, the typicality of the artefacts as classic puns, and the intentionality by providing the connection between the setup and punchline. This paper proposes the PAUL BOT system that incorporates the JAPE system to create puns in order to produce a humorous, creative system utilized for entertainment.

Source: github.com/knw7x9/PunGenerator

Introduction

The most successful joke-generation systems have concentrated on pun generation. The JAPE system, “Joke Analysis and Production Engine”, exemplifies a humorous system that generates a wide range of puns that are consistently evaluated as novel and valuable (Ritchie 2003). An improved version of JAPE has since been developed known as the STANDUP system, “System To Augment Non-speakers’ Dialog Using Puns” that teaches children with communication impairments to tell novel jokes (Waller et al. 2009).

The purpose of JAPE is to produce short texts that are intended to be punning riddles. The fundamental aspects of JAPE are the schemata, sentence forms, templates, and the SAD generation rules (Ritchie 2003). JAPE uses lexemes from riddles, which are phrases that contain linguistic information about a verb, adjective, or noun. Schemata identify the configuration of the lexemes in various riddles. The SAD generator, small adequate description, uses the information provided by the schema to construct abstract linguistic structures called SADs from the lexemes. The SAD generator follows various SAD rules that satisfy preconditions for generating linguistic data according to the information provided by the schema. The relations between the lexemes and the derived constituents, called SAD relations, are transferred to the template stage. The template matcher chooses sentence forms where the conditions are satisfied for insertion. Lastly, the grammar rule generator inserts the parameters into slots within the fixed text to produce the punning riddle.

JAPE incorporates many schemata because there are different lexical preconditions that are possible for each variation of comparable strings. Jokes typically have the same sentence structure, but because of the grammatically diverse English language separate schemata are used.

We have designed a computational humorous system called PAUL BOT, “pun are usually lame but occasionally terrific”. Though similar in many respects to JAPE, PAUL BOT leverages several more recent knowledge bases, including The Corpus of Contemporary American English, ConceptNet, and the CMUdict, all of which we discuss below. Our main interest of PAUL BOT is to create puns that users find humorous in some way. In this paper, we focus on the design of the PAUL BOT system and the analysis of the artefacts produced.

Much of the design of PAUL BOT relies on previous work in the joke generation field. Our system filters a two-word database for adjective-noun phrases which circumvents the need for identifying the linguistic information about a particular phrase in schema.

Similarly to JAPE’s SAD generator, the system constructs abstract linguistic structures from the two-word phrase (Ritchie 2003). The punchline to the joke replaces a word in the original two-word phrase with a homophone via the 2-Gram Database from Corpus of Contemporary American English (COCA) (Weide 2005). The setup to the pun uses WordNet to identify relationships between words on the basis of synonyms, antonyms, and hypernyms (Fellbaum 2012).
Finally, PAUL BOT chooses a template depending on whether the synonym/antonym or hypernym is a verb and whether the sentence structure should be negated for an antonym. Like the template matcher in JAPE, these conditions need to be satisfied in order to choose the correct format of the question (Ritchie 2003). Then, the synonym/antonym and hypernym are inserted into the slots within the chosen template and the appropriate article is added to the noun, which is comparable to the grammar rule generator.

**Methods**

In this section we describe the design and operation of the PAUL BOT system. We first provide a high-level overview of the system. Then, the two-word database, homophone dictionary, synonyms, antonyms, hypernyms, and templates are explored further in detail. Finally, we define the metrics for our user evaluation and special package installations for Python.

**PAUL BOT System Overview**

Taking as input a two-word phrase, the system transforms the phrase using semantic relationships with other words to output a humorous pun. In this case, we define the generated artifact as question-answer pair, where the question acts as a prompt, and the answer is an altered two-word phrase which contains the pun. The flow of information through PAUL BOT is as follows, Figure 1.

1. Choose a two-word phrase at random from a database of adjective-noun phrases as the input into the system.
2. Randomly select one of the two words within the two-word phrase and identify a homophone for this word.
3. Replace the selected word with the homophone found in the previous step. This forms the answer portion of the artifact.
4. Identify a synonym or antonym for the substituted homophone.
5. For the word in the original phrase that was not replaced by a homophone, identify a hypernym.
6. Select the appropriate question template that matches the parts of speech of the synonym or antonym and the hypernym chosen in the previous two steps.
7. Insert the synonym or antonym and the hypernym into the question template.
8. Output the generated pun.

The following demonstrates an example of the flow of information through the system. The chosen two-word phrase input is *electric motor*. A homophone for motor is *voter*. A synonym for voter is *elector*. A hypernym for motor is *car* through a part-of relationship. The output is

What do you get when you cross a car with an elector? *electric voter*

**Two-word Database Synopsis**

A 2-Gram database of two-word phrases was obtained from COCA, Corpus Of Contemporary American English (Davies 2014). The database included the two words, their associated parts of speech, and the frequency of the phrase in the English language. The punchline answer to jokes generated by PAUL BOT is always an adjective-noun combination of words, so we filtered the database for phrases that only contained an adjective followed by a noun.

**Homophone Examination**

The CMUdict, a dictionary of homophones, was retrieved from Carnegie Mellon University (Weide 2005). The dic-
tionary is composed of words and their phonetic spellings. The Levenshtein distance was used to measure the minimum number of single-character edits (insertions, deletions, or substitutions) to change one word into the other. Our algorithm used the Levenshtein distance to measure the difference between the phonetic spellings of two words. If a word is within one edit distance of the search word, this word is considered a homophone. The homophone with the highest frequency in the English language is chosen via the wordfreq package.

**Synonyms, Antonyms, and Hypernyms Overview**

WordNet was utilized for grouping words by semantic relations for synonyms, antonyms, and hypernyms (Fellbaum 2012). The synonyms and antonyms are found in the lemmas of synsets which are data elements considered to be semantically equivalent. Our algorithm uses the wordnet module of the nltk package to decipher synonyms and antonyms of a word. WordNet was used for obtaining type-of relationships for hypernyms. The hypernym and synonym or antonym with the highest frequency in the English language is chosen.

**Template Identification**

The template is chosen depending on two factors: the parts of speech and the negation. The hypernyms, synonyms, or antonyms must be either an adjective, noun, or verb. If one of these is a verb, the “What do you _ that is _?” template is used. Otherwise, the chosen template is “What do you get when you cross _ with _?” The negation of these templates is included to accommodate antonyms.

**Evaluation Metrics**

The PAUL BOT system was run consecutively to produce 150 artefacts. From these artefacts, we manually selected what we deemed the 10 best artefacts for survey evaluation. Ten users evaluated the puns on a scale of 1 to 5 on their funniness, surprise, cleverness, did the user laugh, wit, ingenuity, timelessness, and accessibility.

**Installation of Packages**

Our system utilizes the Levenshtein, nltk, and wordfreq packages. The Levenshtein package measures the edit distance between the phonetic spellings of homophones. The nltk package provides access to WordNet to determine synonyms and antonyms. Lastly, the wordfreq package provides the frequencies of words in the English language.

**Results**

A survey of jokes was administered to ten participants to assess the creativity of the PAUL BOT system. Ten jokes were rated on a scale of 1 to 5 on their funniness, surprise, cleverness, if the user laughed, wit, ingenuity, timelessness, and accessibility on a scale of 1 to 5.

<table>
<thead>
<tr>
<th>Average Rating of Jokes</th>
</tr>
</thead>
<tbody>
<tr>
<td>What do you get when you cross a people with a fire? Human burning</td>
</tr>
<tr>
<td>What do you get when you cross a time with a self-will? Temporary possession</td>
</tr>
<tr>
<td>What do you get when you cross a mother with a sign? Uterine signing</td>
</tr>
<tr>
<td>What do you get when you cross a living with a jerk? Utterine signing</td>
</tr>
<tr>
<td>What do you get when you cross a training with a predator? Corporate vultures</td>
</tr>
<tr>
<td>What do you get when you cross a butter with a cycle? Buttered bikeing</td>
</tr>
<tr>
<td>What do you have that is bad? Tough sex</td>
</tr>
<tr>
<td>What do you create that is big? Large shit</td>
</tr>
<tr>
<td>What do you get when you cross a woman with full? Wide women</td>
</tr>
</tbody>
</table>

The average evaluation of the artefacts across all the features is shown in Table 1. The overall evaluations of the puns range from 2.21 to 3.30. The highest rated pun was “What do you get when you cross a training with a predator? Corporate vultures” with an average rating of 3.3 / 5.0. The lowest evaluated pun with an average evaluation of 2.2 / 5.0 was “What do you get when you cross a woman with full? wide women.”

The median evaluation of the puns according to funniness, surprise, cleverness, if the user laughed, wit, ingenuity, timelessness, and accessibility is shown in Figure 2. Out of all the features of a joke, the artefacts were the most surprising with a 3.35/5.00 evaluation. Unfortunately, the artefacts performed the worst in making the user laugh with a median of 2.0/5.0 and an average of 2.6/5.0. The rest of the features of a joke performed marginally above the midpoint with a rating of 3.0/5.0.

According to Table 1, the highest rated pun was, “What do you get when you cross a training with a predator? corporate vultures.” As shown in Figure 3, users considered this pun to be clever, witty, and accessible with a median evaluation of 4.0 / 5.0. The average user thought this pun was funny with a 3.5/5.0 evaluation. However, this pun did not make the user laugh with an overall 2.0/5.0 rating.
Discussion and Conclusion

In this paper, we explored the design of the PAUL BOT system and the analysis of the artefacts produced. We discovered that our puns usually did not make the user laugh. However, the users were usually surprised by the joke. The funniness, cleverness, wit, ingenuity, timelessness, and accessibility of the pun were marginally better than midpoint.

We believe that PAUL BOT is a computational creative system that exemplifies novelty, value, typicality, and surprise with limited intentionality. PAUL BOT exhibits P-creativity through its creation of puns without a prior knowledge base of existing puns (Wiggins 2006). The artefacts produced are unique due to the transformation of words and therefore novel. The worth or value of the artefacts is moderate because users found the puns to be marginally funny. Our system has high typicality because the artefacts produced by our system represent an ordinary, classic pun. Our system has limited intentionality because the artefact is framed by showing the connection between the setup and the punchline shown in Figure 1. Our puns have a high level of surprise as shown in Figure 2. Because of these points, we believe the PAUL BOT system meets the criteria for generalization (Ventura 2016). Our system does have a fitness function for generating a pun. The word with the highest frequency is chosen from various word lists including homophones, hypernyms, and synonyms/antonyms. However, there is no fitness function for choosing the best puns that are above a certain threshold. Therefore, we argue that PAUL BOT is in between the generalization and filtration stages on a scale of being merely generative to being creative.

Some of the remaining issues that need to be addressed in PAUL BOT are the limited diversity of question templates, failure to appropriately add articles to nouns in generated artifacts, inclusion of unsolicited adult content, and failure to incorporate homonyms. Our system is limited to only having four templates for the question where the hypernym and synonym/antonym can be inserted into the slots. For example, the lowest-rated joke in Table 1 could be improved by adding a template in the following format, “What do you call a woman that is full? wide women.” Our system also has loose constraints on adding an article to a noun by misplacing an article on a plural noun, e.g. a people. The breath of topics of our puns narrows the audience to only adults. Children are not advised to use PAUL BOT. Our system currently does not support the use of homonyms, words that have the same spelling or pronunciation but different meanings, only homophones. As seen in Figure 2, our system cannot produce a pun that is liked by every user.

Despite these limitations, PAUL BOT will be developed further to provide humor for adults. Our next steps are to accommodate more templates, refine the algorithm for adding articles to nouns, and incorporate homonym choices. Our goal is to create a computational humorous system that is capable of making users laugh.

Acknowledgments

Credit goes to Andrew Christiansen and Andres Sewell for helping us to devise the method of hypernym selection within our system.

References

Davies, M. 2014. N-grams data from the corpus of contemporary american english (coca).

Figure 3: The best joke evaluation according to funniness, surprise, cleverness, if the user laughed, wit, ingenuity, timelessness, and accessibility on a scale of 1 to 5.
TECo: Exploring Word Embeddings for Text Adaptation to a given Context

Rui Mendes & Hugo Gonçalo Oliveira
CISUC, Department of Informatics Engineering
University of Coimbra, Portugal
rppm@student.dei.uc.pt, hroliv@dei.uc.pt

Abstract
TECo adapts well-known sayings (e.g., proverbs, movie titles) for a context, given by a textual input (e.g. news headline). For this, it may use one of three methods – word substitution, analogy and vector difference – which are different ways of exploiting word embeddings for word replacement towards a new text that should be semantically related to the input, in the sense that it could be used as a more creative sub-title or comment. These were further combined with two selection methods, based on word overlap and on sentence embeddings, and used in the production of text in context with a small set of Portuguese headlines. To better understand how well our purpose was suit, results were manually assessed. All methods produced text with both syntax and relatedness to the input above average, contrasting the underachieving funniness scores.

Introduction
To amplify the range of a given story, either real or made-up, authors commonly reuse expressions or sayings known by a general audience as a title or subtitle, sometimes also achieving a humorous effect. If the saying is related enough, it can be used directly, but it may also suffer minor adaptations, to become more related to the context and still resemble the original saying. Working on the automation of this process is thus natural. In fact, in scope of linguistic computational creativity, related systems have been developed for generating new creative headlines by resorting to figurative language (Alhajjar, Leppänen, and Toivonen 2019), or blending them with well-known expressions (Gatti et al. 2015); poetry inspired by news stories (Colton, Goodwin, and Veale 2012; Chrismartin and Manurung 2015) or Twitter trends (Gonçalo Oliveira 2017); or applying metaphors to the current news (Veale, Chen, and Li 2017). Having in mind the transformation of text with replacements constrained by the given intentions, operators on word embeddings were even formalised (Bay, Bodily, and Ventura 2019) or applying metaphors to the current news (Veale, Chen, and Li 2017). Other systems simply recommend quotes to be used in dialogues (Ahn et al. 2016), or assign proverbs to news headlines (Mendes and Gonçalo Oliveira 2020).

As a complement to both previous works, and following the idea of using word embeddings, we propose three different methods that exploit this kind of word representation for adapting selected text, so that relatedness to the context increases: substitution of a word by another from or related to the context; substitution of two words by two analogously-related, one of which from the context; and substitution of two words related to the context, in such a way that their relation is preserved.

Although the proposed methods are language-dependent, this study is focused on Portuguese. We rely on Portuguese sayings, namely proverbs and movie titles, and integrate everything in a system dubbed TECo (standing for Texto Em Contexto, in English, Text in Context). Methods based on Term Frequency (TF-IDF) and on sentence embeddings – BERT encodings (Devlin et al. 2019) – were used for both selecting an initial set of sayings to be adapted, and selecting the final result to exhibit, out of all produced.

To better understand the potential of TECo, we ran different combinations of selection and adaptation methods and assessed their results for 30 headlines, manually. For all methods, syntax was generally good, relatedness was above average and funniness below. The most similar sayings in the original list, according to the selection methods, were assessed with the same criteria, with TF-IDF having comparable scores but BERT clearly lower. This suggests that the proposed adaptation methods are capable of creating new text, from a lower amount of original examples, and of comparable quality.

The paper is organised as follows: after this introduction, the proposed methods are described; results of the manual assessment are then presented and discussed; finally, we conclude with a brief discussion.

Methodology
The goal of TECo is to produce new text that resembles a known saying but is related, as much as possible, to an input short text, such that it can be used as a more creative way of transmitting the same idea, complementing it or just commenting on it. We propose three automatic methods for this adaptation: Substitution, Analogy and Vector Difference (hereafter, VecDiff). Besides a set of well-known sayings (TECo’s knowledge base, hereafter KB), to be modified according to the input text (in this case, news headlines), all methods: (i) exploit a pre-trained model of static word embeddings, where words are represented by dense numeric vectors; (ii) assume that the most relevant words in a text are
the open-class words (nouns, verbs, adjectives) that are used in a large corpus but have the lowest frequency (roughly, a high Inverse Document Frequency); (iii) go through all the sayings in a set and try to make adaptations focused on the most relevant words of both the sayings and the input texts. Methods only differ on the adopted strategies for selecting the word(s) to replace.

The first method, Substitution, is the simplest. It replaces the most relevant word in the saying, $a$, by a word from the input text, $b$. Our intuition is that, by using a relevant word of the input text, the meaning of the saying becomes more semantically-related to the given context.

The second method, Analogy, relies on a common operation for computing analogies in word embeddings, i.e., $b - a + a^* = b^*$ (Mikolov, Yih, and Zweig 2013), phrased as $b^*$ is to $b$ as $a^*$ is to $a$. The strategy is to use the two most relevant words in the saying as $a$ and $a^*$, and the most relevant word in the input as $b$. Then: (i) from the previous three, compute a new word $b^*$; (ii) in the original saying, replace $a$ and $a^*$, respectively by $b$ and $b^*$. Given that both pairs of words are analogously-related, our intuition is that the result will still make sense and be more related to the input text.

The third method, VecDiff, also selects the two most relevant words in the input text, $b$ and $b^*$, and then: (i) computes the vector between the previous $b - b^*$; (ii) identifies the pair of open-class words in the saying, $a$ and $a^*$, such that $a - a^*$ maximises the (cosine) similarity with $b - b^*$; (iii) replace $a$ and $a^*$ respectively by $b$ and $b^*$. Our intuition is that the new text will not only use two words of the input, and thus be more related, but also that they will be included in such a way that their relation is roughly preserved.

Although the proposed methods are language-dependent, TECo is focused on Portuguese, our mother tongue. Its KB includes 1,600 Portuguese proverbs from project Natur$^1$ and over 3,000 movie titles in Portuguese, from IMDB$^2$. Most should be well-known, as proverbs are part of the quotididan of most Portuguese people, being used to emphasize certain situations, usually implying some kind of humour. Moreover, these sayings are not usually to be taken literally, as they use several stylistic variations and their underlying meaning may not be clearly understood by a computer.

Table 1 illustrates some results of each method in this context, including an original headline, a proverb and the resulting output. Replaced words and their replacements are underlined. Results were produced with a pre-trained GloVe model of word embeddings, with 300-sized vectors (Hartmann et al. 2017), and relevant words were computed with the help of the newspaper corpus CETEMPúblico (Rocha and Santos 2000). In the first example, $b = \text{bancos}$ replaces $a = \text{amigos}$. In the second, $a = \text{deixes}$, $a^* = \text{fazer}$ and $b = \text{apontar}$, with $b^* = b - a + a^* = \text{comeces}$. In the final example, $a = \text{fere}$, $a^* = \text{ferido}$, to which, out of the words in the headline, $b = \text{finge}$ and $b^* = \text{detido}$ is the pair with the most similar difference.

To avoid syntactic inconsistencies, for any method, replacement candidates must match the morphology of the replaced word, including part-of-speech (PoS) and gender and number, obtained from the morphology lexicon LABEL-Lex (Ranchhod, Mota, and Baptista 1999). If necessary, it may suffer a disambiguation process with a PoS tagger, for which we used the one in NLPyPort (Ferreira, Goncalo Oliveira, and Rodrigues 2019). The latter is also used for lemmatization enabling that, if morphology does not match, the lemma of the candidate can be inflicted to the target form, with the help of the lexicon. If it is still not possible, the saying is just not considered. In any case, the set of possible replacements can be augmented by considering not only the relevant words in the input text, but also the most semantically-similar words, computed in the embeddings. For example, in the Substitution method, $a$ can be replaced by a word different but semantically similar to $b$.

Finally, running through all sayings in the KB should result in several new texts. Even if, due to the morphology constraints, some sayings end up not being used, if similar words are considered for the same input, the same method may produce several variations of the same text. Therefore, a final step has to select the most similar text with the input, according to a sentence similarity method, such as those previously tested in a similar scenario (Mendes and Gonçalo Oliveira 2020). Also, to avoid that, for each input, all sayings in the KB are tested, an initial selection may also rely on such similarity methods or their combination.

**Evaluation**

To take initial conclusions, we ran all the methods in a set of 30 news headlines, with results of each method then manually assessed by two human judges. Besides some insights on the suitability of each method for our purpose, we tested

---

1. https://natura.di.uminho.pt
2. https://www.imdb.com/interfaces/
combinations of two selection methods – TF-IDF and BERT – and included original sayings, directly selected by each of them, in the evaluation. Both get the most similar sayings to the input context. The difference is that TF-IDF represents each text as a weighted vector, based on all the sayings, while BERT encodes each text as a 768-sized vector according to pre-trained model covering 104 languages. Those methods were also used before and after adaptation, for making the initial selection of sayings to use, and for selecting the final output, out of all adapted sayings. Therefore, a total of 14 texts were obtained for each headline: twelve (3×(2+2)) produced by each method with a different combination of selection methods, plus two by each selection method alone. The initial selection contained 100 sayings, including the top-50 related according to the selection method and a random selection of other 50 sayings, for higher diversity.

Judges were presented with headlines, followed by the list of texts by each combination, and were asked to use a 3-point Likert scale for ranking: syntax (1, text has several grammatical and/or structural issues, and may be difficult to interpret; 2, text has minor issues regarding grammar and structure, but is still understandable; 3, text does not have any grammatical or structural issues); relatedness (1, minimal or no relation at all between the text and the headline; 2, text somewhat related to the input; 3, relation between the text and the headline is clear / could be used as a substitute or a comment); and funniness (1, not funny and will not make anyone laugh; 2, somewhat funny and could be potentially funny, depending on the reader’s subjective view; 3, very funny, with a great potential to make people laugh). Table 2 shows the distribution of scores and their median (Md), for texts produced by each adaptation method, regardless of the judge and selection methods, plus the output of the two selection methods, when applied directly to the full KB.

Judge agreement, measured with Cohen’s Kappa, was 0.57 (moderate) for syntax, 0.35 (fair) for relatedness and 0.17 (fair) for funniness. Syntax is more objective, and thus agreement was higher. On the other hand, the other two aspects, especially funniness, are highly subjective, also due to the structure and figurative language of the Portuguese proverbs and vagueness of some movie titles.

According to the scores, syntax is not severely affected by the adaptations, meaning that the produced text is generally grammatical. Few exceptions occur in the adaptation of verbs. Specifically, in Portuguese, the same verb has often different forms for different tenses, genders and numbers, but the same form may also work for different tenses. Thus, incorrectly identifying the tense in the original saying may result in using an incorrect form in the adapted text.

Regarding relatedness, scores are above average. We highlight VecDiff with 64% texts clearly related with the headline and only 13% with no relation. Analogy got 50% clearly related and Substitution 42%, with 35% not related. This makes sense because, while Substitution makes a single replacement, the other two replace two words that, nevertheless, try to keep the original relation, with VecDiff using two words from the original context. On the selection methods, TF-IDF surprisingly got 61% clearly related selections, but BERT selected the lowest proportion of related sayings.

On funniness, results were not as good, as very funny texts were not much more than 10%. This may be due to the aforementioned subjectivity of humour, which may hamper the judge’s decision to give the maximum funniness to a text, because actually making other people laugh depends on many variables. Moreover, the capability of producing content with humoristic value is highly dependent on the context of the input, e.g. it is harder to generate content regarding sad news headlines.

Table 3 illustrates some of the results produced. The first three got the maximum score in all aspects by all judges, and the final two got the lowest scores in relatedness and funniness. Furthermore, it is important to state that most of the resulting texts with minimum scores in both relatedness and funniness used BERT for their final selection, as it seems to suffer from the figurative language used in the sayings, and often selects one that is too distant from the headline, with less focus on shared words. On the other hand, TF-IDF tends to select expressions that share words with the input, thus increasing their relatedness and achieving scores similar to the adaptation methods. This should, however, be analysed more deeply in the future.

Conclusion

Briefly, this study proposed three text adaptation methods to bring a well-known saying closer to a given context, with positive results on syntax and relatedness to the context, but not so much on funniness. When compared to the usage of existing sayings, selected with TF-IDF, with no adaptation, scores are very similar. This shows that the adaptation methods are indeed capable of creating new syntactically-correct and related text, and thus a good option when the number of sayings is limited. We recall that selection methods were applied to the full KB, with 4,600 sayings, while adaptation used only a subset of 100, which they were able to adapt for increased relatedness.

For future endeavours, it would be prolific to test and as-
sess social reactions to the produced text. The proposed methods could be integrated in a chatbot, as a possible conversational aid, or as a creativity booster, in areas like journalism. In the meantime, TECo is working as a Twitter bot, @TextoEmContexto⁴ that regularly reads the headlines of Portuguese newspapers and posts resulting text, as the example in Figure 1.

Figure 1: Example of a tweet posted by the Twitter bot.

Acknowledgements: This work was partially supported by FCT’s INCoDe 2030 initiative, in the scope of the demonstration project AIA, “Apoio Inteligente a Empreendedores (chatbots)”.

Table 3: Examples of produced texts, along with their adaptation and selection methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Headline</th>
<th>Proverb</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Substit + TF-IDF</td>
<td>Rooney sobre cortes de salários: ‘Porque é que são os futebolistas os bodes expiatórios?’ (Rooney about salary cuts: ‘Why are footballers the scapegoats?)</td>
<td>Ós amigos são para as ocasiões (Friends are for the occasions)</td>
<td>Ós expiatórios são para as ocasiões. (Scapegoats are for the occasions)</td>
</tr>
<tr>
<td>Analogy + TF-IDF</td>
<td>Bancos dizem que as condições das linhas de crédito foram definidas pelo governo (Banks claim that credit conditions were defined by the government)</td>
<td>paga o justo pelo pecador (The fair pays for the sinner)</td>
<td>paga o definido pelo pecador (The defined pays for the sinner)</td>
</tr>
<tr>
<td>VecDiff + BERT</td>
<td>Ronaldo junta a família na quarentena para cantar os parabéns à sobrinha (Ronaldo brings family together during quarantine to sing happy birthday to his niece)</td>
<td>papagaio come o milho, periquito leva a fama. (Parrot eats the corn, but the parakeet gets the fame)</td>
<td>papagaio come o milho, sobrinhinho leva a fama. (Parrot eats the corn, but the little nephew gets the fame)</td>
</tr>
<tr>
<td>Substit + BERT</td>
<td>Finge ter Covid-19 no Facebook e acaba detido (Pretends to have Covid-19 on Facebook and ends up arrested)</td>
<td>Nem por ser Natal (Not even for being Christmas)</td>
<td>nem por ter natal (Not even for having Christmas)</td>
</tr>
<tr>
<td>Substit + BERT</td>
<td>Trabalhadores da hotelaria e turismo há quase dois meses sem salários (Workers from hotels and tourism have been without salary for two months)</td>
<td>Mãe só há uma (There is only one mother)</td>
<td>Semana só há uma (There is only one week)</td>
</tr>
</tbody>
</table>

References


4. Music and Poetry
Drum Beats and Where To Find Them: Sampling Drum Patterns from a Latent Space

Alexey Tikhonov
Yandex,
Berlin, Germany
alttsoph@gmail.com

Ivan P. Yamshchikov
Max Planck Institute
for Mathematics in the Sciences,
Leipzig, Germany
ivan@yamshchikov.info

Abstract
This paper presents a large dataset of drum patterns and compares two different architectures of artificial neural networks that produce latent explorable spaces with some recognizable genre areas. Adversarially constrained autoencoder interpolations (ACAI) show better results in comparison with a standard variational autoencoder. To our knowledge, this is the first application of ACAI to drum-pattern generation.

Introduction
In recent years, there have been many projects dedicated to neural network-generated music. For an extended survey of such methods see (Briot, Hadjeres, and Pachet 2019). Some of them are dedicated to drum patterns in particular, however there were several attempts to automate the process of music composition long before the era of artificial neural networks. The well-developed theory of music inspired many heuristic approaches to automated music composition. The earliest idea that we know of dates as far back as the nineteenth century, see (Lovelace 1843). In the middle of the twentieth century, a Markov chain approach for music composition was developed in (Hiller and Isaacson 1959). (Lin and Tegmark 2017) have demonstrated that music, as well as some other types of human-generated discrete time series, tends to have long-distance dependencies that cannot be captured by models based on Markov-chains. Recurrent neural networks (RNNs) seem to be better at processing data series with longer internal dependencies (Sundermeyer, Schlüter, and Ney 2015), such as sequences of notes in tune, see (Boulanger-Lewandowski, Bengio, and Vincent 2012).

Indeed, a variety of different recurrent neural networks such as hierarchical RNN, gated RNN, Long-Short Term Memory (LSTM) network, Recurrent Highway Network, etc., were successfully used for music generation in (Chu, Urtasun, and Fidler 2016), (Colombo et al. 2016), (Johnson 2017), (Yamshchikov and Tikhonov 2017). Google Magenta released a series of projects dedicated to music generation. In particular, one should mention a music_vae model (Roberts et al. 2018) that could be regarded as an extension of drum_rnn\(^1\). It is important to distinguish the generative models like music_vae and the generative models for music that use a straightforward language model approach and predict the next sound using the previous one as an input. For example, (Choi, Fazekas, and Sandler 2016) used a language model approach to predict the next step in a beat with an LSTM. Variational autoencoder (VAE), see (Bowman et al. 2016) and (Semeniuta, Severny, and Barth 2017), on the other hand, allows us to construct a latent space in which each point corresponds to a melody. Such spaces obtained with VAE or any other suitable architecture are of particular interest for different tasks connected with computational creativity since they can be used both to study and classify musical structures, as well as to generate new tunes with specified characteristics.

In this paper, we construct a latent explorable drum pattern space with some recognizable genre areas. Two different smoothing methods are used on the latent space of representations. The obtained latent space is then used to sample new patterns. We experiment with two techniques, namely, variational autoencoder and adversarially constrained autoencoder interpolations (ACAI) (Berthelot et al. 2018).

The contribution of this paper is three-fold: (1) we publish a large dataset of drum patterns, (2) develop an overall representation of typical beat patterns mapped into a two-dimensional space, and (3) compare two different architectures of artificial neural networks that produce explorable spaces of latent representations and demonstrate that VAE seems to produce space with better geometric interpretability that allocates tacks of similar genres closer to each other, yet this does not necessarily correspond to a better subjective quality of the generated samples. ACAI is shown to outperform VAE in terms of the entropy-based quality estimates of the generated percussion patterns as well as in terms of subjective quality assessment.

Dataset
Most of the projects that we know of used small datasets of manually selected and cleaned beat patterns. One should mention a GrooveMonkee free loop pack\(^2\), free drum loops collection\(^3\) and aq-Hip-Hop-Beats-60–110-bpm\(^4\) or (Gillick et al. 2019).

---

\(^1\)https://github.com/tensorflow/magenta/tree/master/magenta

\(^2\)https://groovemonkee.com/collections/midi-loops


\(^4\)https://codepen.io/teropa/details/JLjXGK
Unfortunately, majority of these datasets are either restricted to one or two specific genres or contain very limited amount of midi samples that does not exceed a dozen per genre. This amount of data is not enough to infer a genre-related latent space. Inferring this space, however, could be of utmost importance. Due to the interpolative properties of the model that could work on such space, one can produce infinitely diverse patterns that still adhere to the genre-specific macro-structure. Groove MIDI (Gillick et al. 2019) to a certain extent goes in line with the material presented in the papers yet it is not big enough for the inference of the genre.

Here we introduce a completely new dataset of MIDI drum patterns\(^5\) that we automatically extracted from a vast MIDI collection available online. This dataset is based on approximately two hundred thousand MIDI files, and as we show later is big enough to infer the macroscopic structure of the underlying latent space with unsupervised methods.

Data filtering

The pre-processing of the data was done as follows. Since the ninth channel is associated with percussion according to the MIDI standard, we assumed that we are only interested in the tracks that have non-trivial information in it. All the tracks with trivial ninth channels were filtered out. This filtering left us with almost ninety thousand tracks. Additional filtering included an application of a 4/4 time signature and quantization of the tracks. We are aware that such pre-processing is coarse since it ultimately corrupts several relatively popular rhythmic structures, for example, waltzes, yet the vast majority of the rhythmic patterns are still non-trivial after such pre-processing. We believe that 4/4 time signature is not a prerequisite for the reproduction of the results demonstrated here and encourage researchers to experiment and publish broad and diverse datasets of percussion patterns. In order to reduce the dimensionality of the problem, we have simplified the subset of instruments merging the signals from similar instruments. For example, all snares are merged into one snare sound, low and mid-toms into a low tom, whereas and high tom and high mid-tom into a high tom. Finally, we had split the percussion tracks into percussion patterns. Every track was split into separate chunks based on long pauses. If a percussion pattern that was thirty-two steps long occurred at least three times in a row, it was added to the list of viable patterns. Trivial patterns with entropy below a certain minimal threshold were discarded from the list of viable patterns. Finally, every pattern was checked to be unique in all its possible phase shifts. The resulting dataset includes thirty-three thousand of unique patterns in the collection and is published alongside this paper which is an order of magnitude larger that midi available data sources.

Data representation

The resulting dataset consists of similarly structured percussion patterns. Each pattern has thirty-two time ticks for

\(^5\)https://github.com/altosph/drum_space/blob/master/dataset.tsv

| // Filtering original MIDI dataset |
|-------------------|-------------------|
| for new_track in MIDI_dataset do |
| if new_track[9th_channel] is non-trivial |
| // Quantize with 4/4 signature |
| drum_track ← new_track[9th_channel].quantize() |
| // Merge different drums according to a predefined table |
| drum_track.merge_drums() |
| // Split drum track into chunks |
| for new_chunk in drum_track.split_by_pauses() do |
| if length(new_chunk) == 32 |
| and new_chunk \( \in \) drum_track |
| and entropy(new_chunk) > k |
| percussion_patterns.append(new_chunk) |

| // Filtering non-unique percussion patterns |
|-------------------|-------------------|
| for new_pattern in percussion_patterns do |
| // Create all possible shifts of a pattern |
| shifted_patterns ← new_pattern.all_shifts() |
| // Search for patterns that duplicate and delete them |
| for pattern in percussion_patterns do |
| if pattern ∈ shifted_patterns |
| delete pattern |
| [new_pattern] + percussion_patterns |

Table 1: Pseudo-code that describes filtering heuristics used to form the dataset of percussion patterns.

fourteen possible percussion instruments left after the simplification. Each pattern could be represented as a 14 × 32 matrix with ones on the positions, where corresponding instruments makes a hit. Figure 1 shows possible two-dimensional representations of the resulting patterns.

We can also list all possible combinations of fourteen instruments that can play at the same time tick. In this representation, each pattern is described by thirty-two integers in the range from 0 to 16383. Such representation is straightforward and could be convenient for processing the data with modern models used for generation of discrete sequences (think of a generative model with a vocabulary consisting of 2\(^{14}\) words). The dataset final dataset is published in the following format:

- the first column holds the pattern code that consists of thirty-two comma-separated integers in the range of \([0, 16383]\);
- the second column holds four comma-separated float values that represent the point of this pattern in the latent four-dimensional space, that we describe below;
- the third column holds two comma-separated float values of the t-SNE mapping from the four-dimensional latent space into a two dimensional one, see details below.

The model that we describe further works with a two-dimensional representation shown in Figure 1.

Models and experiments

In this papers we experiment with different autoencoders. Let us first briefly clarify the underlying principles of these architectures.
Autoencoders

Autoencoders are a broad class of structures that process the input $x \in \mathbb{R}^d$, through an ‘encoder’ $z = f_\theta(x)$ parametrized by $\theta$ to obtain a latent code $z \in \mathbb{R}^{d_z}$. The latent code is then passed through a ‘decoder’ $\hat{x} = g_\phi(z)$ parametrized by $\phi$ to produce an approximate reconstruction $\hat{x} \in \mathbb{R}^d$ of the input $x$. In this paper $f_\theta$ and $g_\phi$ are multi-layer neural networks. The encoder and decoder are trained simultaneously (i.e. with respect to $\theta$ and $\phi$) to minimize some notion of distance between the input $x$ and the output $\hat{x}$, for example the squared L2 distance $\|x - \hat{x}\|^2$.

Interpolating using an autoencoder describes the process of using the decoder $g_\phi$ to decode a mixture of two latent codes. Typically, the latent codes are combined via a convex combination, so that interpolation amounts to computing $\hat{x}_\alpha = g_\phi(\alpha z_1 + (1 - \alpha) z_2)$ for some $\alpha \in [0, 1]$ where $z_1 = f_\theta(x_1)$ and $z_2 = f_\theta(x_2)$ are the latent codes corresponding to data points $x_1$ and $x_2$. Ideally, adjusting $\alpha$ from 0 to 1 will produce a sequence of realistic datapoints where each subsequent $\hat{x}_\alpha$ is progressively less semantically similar to $x_1$ and more semantically similar to $x_2$. The notion of ‘semantic similarity’ is problem-dependent and ill-defined.

VAE assumes that the data is generated by a directed graphical model $p_\theta(x|h)$ and that the encoder is learning an approximation $q_\phi(h|x)$ to the posterior distribution $p_\theta(h|x)$. This yields an additional loss component and a specific training algorithm called Stochastic Gradient Variational Bayes (SGVB), see (Rezende, Mohamed, and Wierstra 2014) and (Kingma and Welling 2014). The probability distribution of the latent vector of a VAE typically matches that of the training data much closer than a standard autoencoder.

ACAI has different underlying mechanism. It uses a critic network, as is done in Generative Adversarial Networks (GANs) (Goodfellow et al. 2014). The critic is fed interpolations of existing datapoints (i.e. $\hat{x}_\alpha$ as defined above). Its goal is to predict $\alpha$ from $\hat{x}_\alpha$. This could be regarded as a regularization procedure which encourages interpolated outputs to appear more realistic by fooling a critic network which has been trained to recover the mixing coefficient from interpolated data.

Architecture

In this paper, we experiment with a network that consists of a 3-layered fully connected convolutional encoder, and a decoder of the same size. The encoder maps the beat matrix (32*14 bits) into four-dimensional latent space. The first hidden layer has sixty-four neurons; the second one has thirty-two. The ReLU activations are used between the layers, and a sigmoid maps the decoder output back into the bit mask. Figure 2 shows the general architecture of the network.

The crucial part of the model that is valid for further experiments is the space of latent codes or the so-called ‘bottle-neck’ of the architecture shown in Figure 2. This is a four-dimensional space of latent representations $z \in \mathbb{R}^4$. The structural difference between the VAE and ACAI models with which we experiment further occurs exactly in this bottle-neck. The architectures of the encoder $f_\theta$ and decoder $g_\phi$ are equivalent. Effectively, VAE and ACAI could be regarded as two smoothing procedures over the space of latent codes.

Visualization of the obtained latent space

To explore the obtained dataset, we have built an interactive visualization that is available online\(^6\), and is similar to the one described in (Yamshchikov and Tikhonov 2018). This visualization allows us to navigate the resulting latent space of percussion patterns. Training patterns are marked with grey and generated patterns are marked with red. For the interactive visualization, we use a t-SNE projection of the VAE space since it has a more distinctive geometric structure, shown in Figure 3.

Moreover, this visualization, in some sense, validates the data representation proposed above. Indeed, coarsely a third of tracks in the initially collected MIDIs had genre labels in filenames. After training VAE we used these labels to locate and mark the areas with patterns of specific genres. Closely looking at Figure 3 that shows a t-SNE projection of the obtained latent space, one can notice that the geometric clusters in the obtained latent space correspond to the genres of the percussion patterns. The position of the genres on the Figure were determined by the mean of coordinated of the tracks attributed to the corresponding genre. One can see that related genres are closer to each other in the obtained latent space and the overall structure of the space is meaningful. For example the cloud of ‘Rock’ samples is located between ‘Rock’ and ‘Metal’ clouds, whereas ‘Hip-Hop’ is bordering ‘Soul’, ‘Afro’ and ‘Pop’. The fact that VAE managed to capture this correspondence in an unsupervised set up (as a by-product of training with a standard reconstruction loss) demonstrates that chosen data representation is applicable to the proposed task, and the proposed architecture manages to infer a valid latent space of patterns.

\(^6\)http://altsoph.com/pp/dsp/map.html
As we have mentioned above, we compare two different latent space smoothing techniques, namely, VAE and ACAI. It is important to note here that the standard VAE produces results that are good enough: the space mapping is clear and meaningful, as we have mentioned above. At the same time, the ACAI space seems to be smoother, yet harder to visualize in two dimensions.

Figure 4 illustrates this idea, showing the two-dimensional t-SNE mapping of the latent spaces produced by both methods with patterns that correspond to the genre METAL marked with red dots. One can see that ACAI mapping of a particular genre is not as dense as VAE. Due to this reason, we use t-SNE projection of VAE space for the interactive visualization mentioned above and throughout this paper.

However, we argue that the latent space produced with ACAI is better to sample from and discuss it in detail further.

**Generation**

The majority of the auto-encoder based methods generates new samples according to the standard logic. One can sample an arbitrary point from the latent space and use the decoder to convert that point into a new pattern. In the case of VAE one can also narrow the area of sampling and restrict the algorithm in the hope of obtaining beats that would be representative of the style typical for that area. However, an objective metric that could be used for quality estimation of the generated samples is still a matter of discussion. Such objective estimations are even harder in this particular case since the patterns are quantized and consist of thirty-two steps and fourteen instruments. Indeed, virtually any sequence could be a valid percussion pattern, and human evaluation of such tasks is usually costly and, naturally, subjective. We invite the reader to estimate the quality of the generated samples on her own using the demo mentioned above. At the same time we propose a simple heuristical method that allows putting the quality of different architectures into relative perspective.

Table 2 contains pseudo-code that was used for the filtering of the original MIDI dataset. We suggest using per-
Figure 4: The beats from the area that corresponds to the genre metal on the VAE space projection (left) and the ACAI space projection (right). VAE maps the tracks of the same genre closer together and therefore is beneficial for the visualization of the latent space.

<table>
<thead>
<tr>
<th>Model</th>
<th>% of patterns after filtering</th>
</tr>
</thead>
<tbody>
<tr>
<td>AE</td>
<td>28%</td>
</tr>
<tr>
<td>VAE</td>
<td>17%</td>
</tr>
<tr>
<td>ACAI</td>
<td>56%</td>
</tr>
<tr>
<td>Empirical patterns</td>
<td>82%</td>
</tr>
</tbody>
</table>

Table 2: Comparison of the two smoothing methods. ACAI seems to be way more useful for sampling since it produces a valid percussion pattern out of a random point in the latent space more than 50% of the time and is three times more effective than VAE based architecture. In terms of the heuristic entropy filter, VAE performs even worse than AE, generating a lot of “dull” samples with entropy below the threshold.

Discussion

Deep learning enables the rapid development of various generative algorithms. There are various limitations that hinder the arrival of algorithms that could generate discrete sequences that would be indistinguishable from the corresponding sequences generated by humans. In some contexts, the potential of such algorithms might still be limited with the availability of training data; in others, such as natural language, the internal structure of this data might be a challenge; finally, some of such tasks might be simply too intensive computationally and therefore too costly to use. However, percussion patterns do not have such limitations. The structure of the data can be formalized reasonably well and without significant loss of nuance. In this paper, we provide thirty-three thousand thirty-two step 4/4 signature percussion drums and demonstrate that such a dataset allows training a good generative model. We hope that as more and more data is available for experiments, percussion could be the first chapter to be closed in the book of generative music.

Nevertheless, even within the percussive component of music generation, there are a lot of open problems to be solved. For example, there are several works on generative song structure, but they are mostly either heuristically
motivated or anecdotal rather than data-driven. Generative models capable of smooth interpolations between different rhythmic patterns represent another set of new research questions. Finally, nuances of percussion alongside with the datasets and the models that could capture these nuances, for example see (Gillick et al. 2019), need further research.

Conclusion

This paper presents a new huge dataset of MIDI percussion patterns that could be used for further research of generative percussion algorithms.

The paper also explores two autoencoder based architectures that could be successfully trained to generate new MIDI beats. Both structures have similarly fully connected three-layer encoders and decoders but use different methods for smoothing of the produced space of latent representations. Adversarially constrained autoencoder interpolations (ACAI) seem to provide denser representations than the ones produced by a variational autoencoder. More than half of the percussion patterns generated with ACAI passes the simple heuristic filter used as a proxy for the resulting generation quality estimation. To our knowledge, this is the first application of ACAI to drum-pattern generation.

The interactive visualization of the latent space is available as a tool to subjectively assess the quality of the generated percussion patterns.

References


Score and Lyrics-Free Singing Voice Generation

Jen-Yu Liu¹, Yu-Hua Chen¹,², Yin-Cheng Yeh¹ & Yi-Hsuan Yang¹,²
¹Taiwan AI Labs, Taipei, Taiwan
²Academia Sinica, Taipei, Taiwan

jyliu@ailabs.tw, cloud60138@citi.sinica.edu.tw, yyeh@ailabs.tw, yang@citi.sinica.edu.tw

Abstract

Generative models for singing voice have been mostly concerned with the task of “singing voice synthesis,” i.e., to produce singing voice waveforms given musical scores and text lyrics. In this work, we explore a novel yet challenging alternative: singing voice generation without pre-assigned scores and lyrics, in both training and inference time. In particular, we outline three such generation schemes, and propose a pipeline to tackle these new tasks. Moreover, we implement such models using generative adversarial networks and evaluate them both objectively and subjectively.

Introduction

The task of computationally producing singing voices is usually referred to as singing voice synthesis (SVS) in the literature (Cook 1996). Most researchers assume that the note sequence and the lyrics of the audio to be generated are given as the model input, and aim to build synthesis engines that sound as natural and expressive as a real singer (e.g., (Hono et al. 2019; Lee et al. 2019)). As such, the content of the produced singing voice is largely determined by the given model input, which is usually assigned by human.

However, singing according to a pre-assigned musical score and lyrics is only a part of the human singing activities. For example, we learn to spontaneously sing when we were children (Dowling 1984). We do not need a score to sing when we are humming on the road or while taking a shower. The voices sung do not have to be intelligible. Jazz vocalists can improvise according to a chord progression, an accompaniment, or even nothing.

We aim to explore such a new task in this paper: teaching a machine to sing with a training collection of singing voices, but without the corresponding musical scores and lyrics of the training data. Moreover, the machine has to sing without pre-assigned score and lyrics as well even in the inference (generation) time. This task is challenging in that, as the machine sees no lyrics at all, it hardly has any knowledge of the human language to pronounce or articulate either voiced or unvoiced sounds. And, as the machine sees no musical scores at all, it has to find its own way learning the language of music in creating plausible vocal melodies.

Specifically, we consider three types of such score- and lyrics-free singing generation tasks. A free singer sings with only random noises as the input. An accompanied singer learns to sing over a piece of instrumental music, which is given as an audio waveform (again without score information). Finally, a solo singer also sings with only noises as the input, but it uses the noises to firstly generate some kind of ‘inner ideas’ of what to sing. From a technical point of view, we can consider SVS as a strongly conditioned task for generating singing voices, as the target output is well specified by the input. In contrast, the proposed tasks are either unconditioned or weakly conditioned. While our models are presumably more difficult to train than SVS models, they enjoy more freedom in the generation output, which may be desirable considering the artistic nature of singing.

The proposed tasks are challenging in a few aspects.

• First, the tasks are unsupervised as we do not provide any labels (e.g., labels of phonemes or pitches) for the training singing files. The machine has to learn the complex structure of music directly from the audio signals.

• Second, for training the free singer, unaccompanied vocal tracks are needed. As for the accompanied singer, we need additionally an accompaniment track for each vocal track. However, it is hard to amass such multi-track music data from the public domain.

• Third, for the accompanied singer case, there is no single “ground truth” and the relationship between the model input and output may be one-to-many. This is because there are plenty of valid ways to sing over an accompaniment track. For diversity and artistic freedom, we cannot ask the machine to generate any specific singing voice in response to an accompaniment track, even if we have paired data of vocal and accompaniment tracks.

To address the first and third issues, we explore the use of generative adversarial network (GAN), in particular conditional GAN (Mirza and Osindero 2014) to retain the possibility of generating singing voices with multiple modes. Specifically, we design a novel GAN-based architecture to learn to generate the mel-spectrogram of singing voice, and then use a vocoder to generate the audio waveform. Rather than considering the mel-spectrograms as a fixed-size image, we use gated recurrent units (GRUs) and dilated convolutions in both the generator and discriminator, to model both the local and sequential patterns in music and to facilitate the generation of variable-length waveforms.

To address the second issue, we choose to implement a
vocal source separation model with state-of-the-art separation quality (Liu and Yang 2019) for data preparation. The advantage of having a vocal separation model is that we can use as many audio files as we have to compile the training data. The proposed pipeline for training an accompanied singer is illustrated in Figure 1.

We implement the proposed singing voice generation models with a collection of Jazz music. For evaluation, we employ a few objective metrics and conduct a user study. Samples of the generated singing voices can be found at https://bit.ly/2mIvoIc. To demonstrate the use case of using the generated sounds as a sound source, we manually make a song in the style of Jazz Hiphop by sampling the output of a free singer we trained. This song can be heard at https://bit.ly/2QkUJoJ. For reproducibility, we release our code at https://github.com/ciaua/score_lyrics_free_svg.

A free singer takes no conditions at all as the input. We want it to sing freely. The singing voices from a free singer may not even sound good, but they should sound like singing voice. A free singer is like we are freely humming or singing on the road walking or in the bathroom taking a shower. We may not even know what we are singing and likely there is no underlying musical score.

An accompanied singer tries as the input a sequence of accompaniment-derived features. An accompanied singer tries to generate singing voices that match the accompaniment track in some way. It is similar to the case of Karaoke, where a backing accompaniment track is played from a speaker, the lyrics and a video are displayed on a screen, and a user tries to sing according to the lyrics and the backing track. The difference is that, this time the user is a trained model and we do not ask it to follow the lyrics or the exact pitch contour of the accompaniment. The note sequence found in the singing has to be in harmony with, but not a duplicate of, that in the backing track.

A solo singer is similar to a free singer in that both takes no conditions as the input. However, a solo singer would generate an ‘inner idea’ first, and then sing according to that. The inner idea can take several forms. In this work, we instantiate this scheme with the inner idea being a chord progression (namely a sequence of chord labels). The distribution of inner ideas is modeled by an auto-regressive recurrent network we build for chord progression generation. Alternatively, we can think of a solo singer as a combination of an idea generator and an accompanied singer. For an accompanied singer, the information extracted from the given accompaniment track can take several forms such as transcribed pitches and chord progressions. A solo singer learns to generate such information on its own, without reference to an actual accompaniment track.

Models

To account for the absence of supervised data and the highly complicated spatio-temporal patterns in audio spectrograms, we propose a new adversarial net that features heavy use of GRUs (Cho et al. 2014), dilated convolutions (van den Oord et al. 2016), and feature grouping to build our singer models. We provide the algorithmic details below.

Block of GRU-Grouped Dilated Convolution-Group Normalization

Network architectures with stacked blocks of GRUs and dilated convolutions have been used to attain state-of-the-art performance in blind musical source separation (Liu and Yang 2019). In a source separation task, a model learns to decompose, or unmix, different sources (e.g., vocal, piano, bass, drum) from a mixture signal. This requires the abilities to model the relationships between different sources as well as the relationships between neighboring time frames. The output spectrograms are also expected to be distortion-less and of high audio quality. For it has demonstrated its capability in source separation, we adopt it as a building block of the singer models. Especially, we want the singer models to also consider accompaniment information.

Specifically, one such block we adopted in our models is a stack of GRU, dilated convolution with feature grouping, and group normalization (Wu and He 2018). The input to the GRU, the output of the GRU, and the output of the group normalization are summed to form the output of the block. We note that the original ‘D2 block’ used in (Liu and Yang 2019) uses dilated GRU and uses weight normalization (Salimans and Kingma 2016) for the dilated convolution layers. However, empirically we find that it is easier for the singer models to converge by replacing weight normalization with group normalization, and using plain GRUs as good as using dilated GRUs. We refer to our blocks as GRU-grouped dilated convolution-group normalization block (‘G3 block’).

Singer Models with BEGAN, G3 blocks and Frame-wise Noises (G3BEGAN)

The accompanied singers and solo singers have to take conditions as part of their input. One desirable property of the models is the ability to generate voices with arbitrary length,
as the conditional signal can be of variable length. Besides, the model has to deal with the one-to-many issue mentioned in the introduction, and the absence of supervisory signals. With these in mind, we design a GAN architecture for score and lyrics-free voice generation. In particular, we pay special attention to the following three components: 1) the network architecture, 2) the input noises for GAN, and 3) the loss function of the discriminator.

Let us first take a look at two existing GAN models for audio generation: (Engel et al. 2019) and (Donahue, McAuley, and Puckette 2019). Their generators and discriminators are both based on 2D convolutions, transposed 2D convolutions and dense (linear) layers. The generators take a vector \( z \in \mathbb{R}^U \) as the input noise and use transposed convolutions to expand \( z \) so that a temporal dimension emerges in the expanded intermediate matrices. The number of temporal frames in the final output depends on the total strides used in all the transposed convolutions. The discriminators take the output of the generators or the real signal as the input, and compress the input matrix with convolution layers until the output becomes a single value represents the prediction of true (real) or false (generated) data.

A main reason why existing models cannot generate variable-length output is the need to expand \( z \) by transposed convolution layers. We remedy this by using an architecture consisting of the proposed G3 blocks, and convolutions without strides, for both the generators \( G(\cdot) \) and discriminators \( D(\cdot) \). Moreover, instead of using a single noise vector, our models take as input a sequence of noise vectors, denoted as \( Z \in \mathbb{R}^{U \times T} \), that has the same temporal length as the desired output \( Y \). Each column of \( Z \) is sampled independently from a Gaussian distribution \( \text{Normal}(0, 1) \). At the first glance, it might feel unnatural to have one noise vector per frame as that may result in fast oscillations in the noises. However, we note that the output of \( G(\cdot) \) for the \( t \)-th frame depends not only on the \( t \)-th column of \( Z \) (and \( C \) or \( I \)), but the entire \( Z \) (and the condition matrices), due to the recurrent GRUs in the model. We expect that the GRUs in the discriminator \( D(\cdot) \) would force \( G(\cdot) \) to generate consistent consecutive frames. Therefore, the effect of the frame-wise noises might be introducing variations to the generation result, for example by adjusting the modes of the generated frame-wise features.

As for the loss function of \( D(\cdot) \), we experiment with the following three options: the vanilla GAN, the LSGAN (Mao et al. 2017) that adopts the least squares loss function for the discriminator, and the boundary equilibrium GAN (BEGAN) (Berthelot, Schumm, and Metz 2017) that adopts an “auto-encoder style” discriminator loss. The \( D(\cdot) \) in either GAN or LSGAN is implemented as a classifier aiming to distinguish between real and generated samples, whereas the \( D(\cdot) \) in BEGAN is an autoencoder aiming to reconstruct its input. Specifically, in BEGAN, the loss functions \( l_D \) and \( l_G \) for the discriminator and generator, as in the case of the accompanied singer, are respectively:

\[
l_D = L(X, C) - \tau_s L(G(Z, C), C),
\]

\[
l_G = L(G(Z, C), C),
\]

where \( X \in \mathbb{R}^{K \times T} \) is the feature sequence of a real vocal track sampled from the training data, \( G(Z, C) \in \mathbb{R}^{K \times T} \) is the feature sequence for the generated vocal track, and \( L() \) is a function that measures how well the discriminator \( D(\cdot) \), implemented as an auto-encoder, reconstructs its input:

\[
L(M, C) = \frac{1}{WT} \sum_{w,t} |D(M, C)_{w,t} - M_{w,t}|,
\]

for an arbitrary \( W \times T \) matrix \( M \), where we use \( M_{w,t} \) to denote the \((w, t)\)-th element of a matrix \( M \) (and similarly for \( D(M, C)_{w,t} \)). Moreover, the variable \( \tau_s \) in Eq. (1) is introduced by BEGAN to balance the power of \( D(\cdot) \) and \( G(\cdot) \) during the learning process. It is dynamically set to be \( \tau_{s+1} = \tau_s + \lambda (\gamma L(X, C) - L(G(Z, C), C)) \), for each training step \( s \), with \( \tau_s \in [0, 1] \). \( \lambda \) and \( \gamma \) are manually-set hyperparameters.

Our pilot study (not reported here due to space restriction) shows that BEGAN performs the best. Therefore, we consider below the BEGAN-based model, dubbed G3BEGAN. See Table 1 for some details of the network architecture.

### Source Separation (SS)

To get the vocal tracks for training our singer models, we implement a source separation (SS) model following the architecture proposed by (Liu and Yang 2019), which represents the state-of-the-art as evaluated on the MUSDB dataset (Rafi et al. 2017). MUSDB contains clean vocal and accompaniment tracks for various genres. As we are mainly interested in Jazz music in this work, we collect additionally 4.5 hours of Jazz piano solo audio to augment the MUSDB for training our SS model. Our model can isolate out not only the vocal track but also the piano track from an arbitrary song. Please visit \( \text{https://bit.ly/2Xattua} \) for samples of the separation result of our SS model.

### Chord Generator

For training the solo singer, we implement a chord generator in this work. It is aimed to generate chord progres-
sions freely under some given conditions. It supports 12 major and 12 minor keys, 10 tempo options from 60 to 240 BPM, 6 time signature options, and 51 chord qualities (612 chords in total). The conditions, key, tempo, and time signatures, are encoded into one-hot representation and concatenated together as a 40-dimension vector. Our chord generator mainly consists with 3 stacked GRU layers. The input of each time step is a 524-dimensional vector consisting of a chord embedding and a beat-related one-hot positional encoding, to encourage the model to follow certain rhythmical pattern. This input array passes through a fully-connected layer to 512-dimension and is used as the input of the GRU’s. The training data are the lead sheets from the Wikifonia dataset. We augmented the data by rotating the keys, leading to in total 80,040 lead sheets for training.

Experiments

Implementation Details

We use 80-dimensional mel-spectrograms as the acoustic features modeled and generated by the singer models. We use the python package librosa (McFee et al., 2015), with default settings, to compute the mel-spectrograms from audio. A mel-spectrogram is passed to a WaveRNN vocoder (Kalchbrenner et al. 2018) to generate an audio signal from mel-spectrograms. Our implementation of the WaveRNN vocoder is based on the code from Fatchord.1 Instead of using off-the-shelf pre-trained vocoders, which are typically trained for text-to-speech (TTS), we train our vocoder from scratch with a set of 3,500 vocal tracks separated by our SS model from an in-house collection of music that covers diverse musical genres.

We collect 17.4 hours of Jazz songs containing female voices and 7.6 hours of Jazz songs with male voices. We again use our SS model to get the vocal tracks from these songs. For batched training, we divide the tracks into 10-second sub-clips. Sub-clips that contain less than 40% vocals, as measured from energy, are removed. This leads to 9.9-hour and 5.0-hour training data for female and male Jazz vocals, respectively. 200 and 100 sub-clips are reserved from the training set as the validation set for female and male voices, respectively. Singer models with G3BEGAN are trained with Adam (Kingma and Ba, 2014) with $10^{-4}$ learning rate, mini-batch size 5, and gradient norm clipping with magnitude 3. We train the model for 500 epochs, and then pick the epoch with the best convergence rate (Berthelot, Schumm, and Metz, 2017) for evaluation.

For the accompanied singer, we experiment with extracting pitch-related information from the accompaniment track to condition the generation of the vocal track. The assumption here is that whether the generated vocal track is in harmony with the accompaniment track can be largely determined by pitch-related information. For this purpose, we implement a piano transcription model to transcribe the separated piano track, leading to 88-dimensional transcribed frame-wise pitch as the accompaniment condition. We implement our piano transcription model with the G3 blocks, following the training procedure of (Hawthorne et al. 2018). We note that, the clips in the training set of our singer models may not contain piano playing. Even if a clip contains piano playing, the piano may not play across the entire clip. Hence, the models have to learn to sing either with or without the piano accompaniment.

For evaluating the accompanied singer, we collect 5.3 hours of Jazz music from Jamendo.2 As the music hosted on Jamendo are copyright-free, we will later be able to share the test set to the research community. We apply our SS model to the audios to get the piano track, divide each track into 20-second sub-clips,3 and discard those clips that are silent (i.e., do not contain piano). Piano transcription is also applied to the separated piano track, yielding 402 20-second sub-clips for evaluation. As for evaluating the solo singer, we generate 402 chord progressions by our chord generator.

Baselines

As this is a new task, there is no previous work that we can compare with. Therefore, we establish the baselines by 1) computing the baseline objective metrics (see Section ) from the training data of the singing models, and 2) using existing SVS systems for synthesizing singing voices.

For the SVS baselines, we employ Sinsy (Oura et al. 2010; Hono et al. 2018) and Synthesizer V (Hua and others 2019), the two well-known SVS systems that are publicly accessible. For Sinsy, we use the publicly available repository4 to query the Sinsy API;5 we use the HMM version (Oura et al. 2010) instead of the deep learning version as the latter cannot generate male voices. For Synthesizer V, we use their software.6 We use Sinsy for both objective and subjective tests, but Synthesizer V for subjective test only, for the latter does not provide a functionality to batch process a collection of MIDI files and lyrics.

SVS systems require lyrics and melody as the input. For the lyrics, we choose to use multiple ‘la,’ the default lyrics for Synthesizer V.7 For the melodies, we adopt two methods:

- Vocal transcription from singer training data. We use CREPE to transcribe the separated vocals from the singer training data, and convert it to MIDI format.
- Piano transcription from the Jamendo testing data. The transcription result often contains multiple notes at the same time. Hence, we further use the skyline algorithm (Ozcan, Isikhan, and Alpkoçk 2005) to the transcription result to get a melody line comprising the highest notes.

1https://github.com/fatchord/WaveRNN
2https://www.jamendo.com
3Please note that this is longer than the 10-second sub-clips we used to train the singer models. This is okay as our model can generate variable-length output.
4https://github.com/mathigatti/midi2voice
5http://sinsy.jp/
6https://synthesizerv.com/
7As our models do not contain meaningful lyrics, to be fair the baselines should not contain meaningful lyrics either. We choose ‘la’ because people do sometimes sing with ‘la’ and it has no semantic meaning. An alternative way to get the lyrics is by randomly sampling a number of characters. However, randomly sampling a reasonable sequence of characters is not a trivial task.
Proposed model & Average pitch (Hz) & Vocalness & Matchness \\ 
\hline
Free singer (female) & 288 ± 28 & 292 ± 28 & 0.48 ± 0.09 & −13.28 ± 3.80 \\
Accompanied singer (female) & 313 ± 18 & 316 ± 19 & 0.55 ± 0.11 & −9.25 ± 3.13 \\
Solo singer (female) & 302 ± 17 & 306 ± 18 & 0.56 ± 0.10 & −9.30 ± 3.11 \\
\hline
Free singer (male) & 248 ± 39 & 242 ± 32 & 0.44 ± 0.16 & −13.29 ± 3.19 \\
Accompanied singer (male) & 207 ± 14 & 200 ± 15 & 0.44 ± 0.13 & −9.31 ± 3.16 \\
Solo singer (male) & 213 ± 14 & 207 ± 16 & 0.46 ± 0.12 & −9.30 ± 3.13 \\
\hline
Baseline: Singing voice synthesis & & & & \\
Sinsy (training vocal, female) & 305 ± 59 & 308 ± 57 & 0.71 ± 0.17 & −9.20 ± 3.12 \\
Sinsy (training vocal, male) & 260 ± 86 & 259 ± 72 & 0.73 ± 0.14 & −9.09 ± 3.14 \\
Sinsy (testing piano skyline, female) & 523 ± 138 & 431 ± 62 & 0.66 ± 0.14 & −8.88 ± 3.04 \\
Sinsy (testing piano skyline, male) & 520 ± 137 & 423 ± 61 & 0.62 ± 0.15 & −8.93 ± 3.02 \\
\hline
Baseline: Training data & & & & \\
Wikifonia: real melody-chords & — & — & — & −7.04 ± 2.91 \\
Wikifonia: random melody-chords & — & — & — & −13.16 ± 3.72 \\
Singer train data (vocals, female) & 312 ± 70 & 310 ± 56 & 0.60 ± 0.14 & −9.24 ± 3.09 \\
Singer train data (vocals, male) & 263 ± 93 & 258 ± 75 & 0.64 ± 0.16 & −9.09 ± 3.22 \\
Singer train data (accomp., female) & — & — & — & — \\
Singer train data (accomp., male) & — & — & 0.12 ± 0.15 & — \\
MUSDB clean vocals & 271 ± 81 & 283 ± 75 & 0.59 ± 0.14 & — \\
\hline
\end{tabular}

Table 2: Result of objective evaluation for our singer models and a few baseline methods.

Objective Metrics and Objective Evaluation Result

The best way to evaluate the performance of the singer models is to listen to the generated results. Therefore, we encourage readers to listen to the audio files in our demo website, mentioned in the end of the introduction section. However, objective evaluation remains desirable, either for model development or for gaining insights into the generation result. We propose the following metrics for our tasks.

- **Vocalness** measures whether an audio clip contains singing voices. There are different publicly available tools for detecting singing voices in an audio mixture (e.g., (Lee, Choi, and Nam 2018)). We choose the JDC model (Kum and Nam 2019) for it represents the state-of-the-art. In this model, the pitch contour is also predicted in addition to the vocal activation. If the pitch at a frame is outside a reasonable human pitch range (73–988 Hz defined by JDC), the pitch is set to 0 at that frame. We consider a frame as being vocal if it has a vocal activation $\geq 0.5$ AND has a pitch $> 0$. Moreover, we define the vocalness of an audio clip as the proportion of its frames that are vocal. The tool is applied to the non-silent part of an audio\(^8\) of the generated singing voices only, excluding the accompaniment part.

- **Average pitch**: We estimate the pitch (in Hz) for each frame with two pitch detection models: the state-of-the-art monophonic pitch tracker CREPE (Kim et al. 2018a), and JDC. The average pitch is computed by averaging the pitches across the frames with confidence higher than 0.5 for CREPE, and across the frames that are estimated to be vocal for JDC.

- **Singing-accompaniment matchness**: To objectively measure matchness, we build a melody harmonization recurrent network model (MH) by adapting our chord generator, using additionally the melody tracks found in the Wikifonia dataset. Specifically, the MH model intends to generate a chord sequence given a melody sequence. Such a model can be learned by using the pairs of melody and chord tracks in Wikifonia. We add the chroma representation of the melody with window size of a quarter-note to the input vector. Given a pair of melody and chord sequences, the MH model computes the likelihood of observing that chord sequence as the output when taking the melody sequence as the model input. We use the average of the log likelihood across time frames as the matchness score. As the MH model considers symbolic sequences, we use CREPE to transcribe the generated voices, and Madmom (Böck et al. 2016) to recognize the chord sequence from the accompaniment track.

Several observations can be made from the result shown in Table 2. In terms of the average pitch, we can see that the result of our model is fairly close to that of the singing voices in the training data. Moreover, the average pitch of the generated female voices is higher than that of the generated male voices as expected. We can also see that the Sinsy singing voices tend to have overly high pitches, when the melody line is derived from a piano playing (denoted as ‘testing piano skyline.’).

---

\(^8\)The non-silent frames are derived by using the `librosa` function ‘effects_signal_to_frame_nonsilent.’
Table 3: Mean opinion scores (MOS) and standard deviations in four evaluation criteria collected from the first user study, for different versions of accompanied singer (female). The scores are in 5-point Likert scale (1–5); the higher the better.

<table>
<thead>
<tr>
<th>Model (epochs trained)</th>
<th>Sound quality</th>
<th>Vocalness</th>
<th>Expression</th>
<th>Matchness</th>
</tr>
</thead>
<tbody>
<tr>
<td>G3BEGAN (20 epochs)</td>
<td>1.59 ± 0.82</td>
<td>1.93 ± 0.99</td>
<td>1.98 ± 0.88</td>
<td>2.18 ± 1.08</td>
</tr>
<tr>
<td>G3BEGAN (240 epochs)</td>
<td>2.24 ± 0.93</td>
<td>2.66 ± 1.01</td>
<td>2.60 ± 1.01</td>
<td>2.58 ± 1.05</td>
</tr>
<tr>
<td>G3BEGAN (final)</td>
<td>2.38 ± 0.96</td>
<td>2.98 ± 1.02</td>
<td>2.85 ± 1.00</td>
<td>2.74 ± 1.04</td>
</tr>
</tbody>
</table>

Table 4: MOS from the second user study, comparing our model and two existing SVS systems.

<table>
<thead>
<tr>
<th>Model (epochs trained)</th>
<th>Sound quality</th>
<th>Vocalness</th>
<th>Expression</th>
<th>Matchness</th>
</tr>
</thead>
<tbody>
<tr>
<td>G3BEGAN (final)</td>
<td>1.71 ± 0.70</td>
<td>2.39 ± 1.11</td>
<td>2.27 ± 1.06</td>
<td>2.34 ± 1.16</td>
</tr>
<tr>
<td>Sinsy (Oura et al. 2010)</td>
<td>3.19 ± 1.07</td>
<td>2.90 ± 1.01</td>
<td>2.40 ± 0.98</td>
<td>2.10 ± 0.90</td>
</tr>
<tr>
<td>Synthesizer V (Hua and others 2019)</td>
<td>3.57 ± 1.07</td>
<td>3.30 ± 1.24</td>
<td>3.25 ± 1.10</td>
<td>3.35 ± 1.15</td>
</tr>
</tbody>
</table>

In terms of vocalness, our models score in general lower than Sinsy, and the singing voices in the training data. However, the difference is not that far. As a reference, we also compute the vocalness of the accompaniments in the training set (denoted as ‘accomp.’) and it is indeed quite low.9

As for matchness, we show in Table 2 the score computed from the real melody-chords pairs of Wikifonia (~7.04) and that from random pairs of Wikifonia (~13.16). We can see that the accompanied singers score higher than the random baseline and the free singer as expected.10 Moreover, the matchness scores of the accompanied singers are close to that of the singer training data.

From visually inspecting the generated spectrograms and listening to the audio, the models seem to learn the characteristics of the singing melody contour (e.g., the F0 is not stable over time). Moreover, the female singer models learn better than the male counterparts, possibly because of the larger training set.

User Study and Subjective Evaluation Result
We conduct two online user studies to evaluate the accompanied singer, the female one. In the first user study, we compare the ‘final’ model (with the number of epochs selected according to a validation set) against two early versions of the model trained with less epochs. In the second one, we compare the proposed accompanied singer with Sinsy and Synthesizer V.

In the first study, we recruit 39 participants to each rate the generated singing for three different accompaniment tracks (each 20 seconds), one accompaniment track per page. The subjects are informed the purpose of our research (i.e., score and lyrics-free singing voice generation) and the user study (to compare three computer models), and are asked to listen in a quiet environment with proper headphone volume. No post-processing (e.g., noise removal, EQ adjustment) is applied to the audio. The ordering of the result of the three models is randomized.

The process of the second study is similar to the first one, but it includes five different accompaniments (randomly chosen from those used in the first user study) and the respective generated/synthesized singing voices. The melodies used for synthesis are those from the piano skyline of the test data, so that our model can be compared with the synthesis methods with the same accompaniment. A separate set of 21 subjects participate in this study. The audio files used in this user study can be downloaded from https://bit.ly/2qNrekv.

Tables 3 and 4 show the result of the two studies. We can see that the model indeed learns better with more epochs. Among the four evaluation criteria, the Sound Quality is rated lower than the other three in both studies, suggesting room for improvement.

By comparing the proposed model with the two SVS systems, we see that Synthesizer V performs the best for all the evaluation criteria. Our model achieves better Matchness than Sinsy, and achieves a rating close to Sinsy in Expression.11 In general, we consider the result as promising considering that our models are trained from scratch with little knowledge of human language.

Related work
While early work on SVS is mainly based on digital signal processing (DSP) techniques such as sampling concatenation (Cook 1996; Bonada and Serra 2007), machine learning approaches offer greater flexibility and have been more

9We note that Sinsy even scores higher in vocalness than the training data. This may be due to the fact that real singing voices are recorded under different conditions and effects.

10The matchness scores of the free singers are computed by pairing them with the 402 test clips.

11We note that Sinsy and Synthesizer V have an unfair advantage on matchness because their singing voices are basically synthesized according to the melody lines of the accompaniment. From Table 4, we see that Synthesizer V does exhibit this advantage, while Sinsy does not. We observe that the Sinsy singing voices do not always align with the provided scores. The fact that Synthesizer V has higher audio quality seem to promote its score in the other criteria. The presence of the result of Synthesizer V seems to also make the subjects in the second study rate the proposed model lower than the subjects do in the first study.
widely studied in recent years. Hidden Markov models (HMMs), in particular, have been shown to work well for the task (Saino et al. 2006). The Sinsy system, a baseline model in Section , is also based on HMMs (Oura et al. 2010). (Nishimura et al. 2016) report improved naturalness by using deep neural nets instead of HMMs. Since then, many neural network models have been proposed.

The model presented by (Nishimura et al. 2016) uses simple fully-connected layers to map symbolic features extracted from the user-provided scores and lyrics, to a vector of acoustic features for synthesis. The input and output features are time-aligned frame-by-frame beforehand by well-trained HMMs. The input features consist of score-related features (e.g., the key of the current bar and the pitch of the current musical note), and lyrics-related ones (the current phoneme identifier, the number of phonemes in the current syllable, and the duration of the current phoneme). The output features consist of spectral and excitation parameters and their dynamic features (Hono et al. 2018), which altogether can then be turned into audio with a DSP technique called the MLSA filter (Imai 1983).

The aforementioned model has been extended in many aspects. For instance, using convolutional layers and recurrent layers in replacement of the fully-connected layers for learning the mapping between input and output features has been respectively investigated by (Nakamura et al. 2019) and (Kim et al. 2018b). Using neural vocoders such as the WaveNet (van den Oord et al. 2016) instead of the MLSA filter has been shown to improve naturalness by (Nakamura et al. 2019). Rather than using hand-crafted features for the input and output, (Lee et al. 2019) train a model to predict the mel-spectrogram directly from time-aligned lyrics and pitch labels, and then use the Griffin-Lim algorithm (Griffin and Lim 1984) to synthesize the audio. Modern techniques such as adversarial loss and attention module have also been employed (Lee et al. 2019). Synthesizer V (Hua and others 2019), the other baseline model we employ in Section , is based on a hybrid structure that uses both deep learning and sample-based concatenation.\footnote{https://synthv.fandom.com/wiki/File:Synthesizer_V_at_the_Forefront_of_Singing_Synth(last accessed: Nov. 12, 2019)}

While exciting progress has been made to SVS, the case of score and lyrics-free singing voice generation, to our best knowledge, has not been tackled thus far. Similar to (Lee et al. 2019), we do not use hand-crafted features and we train our model to predict the mel-spectrograms.

**Conclusion**

In this paper, we have introduced a novel task of singing voice generation that does not use musical scores and lyrics. Specifically, we proposed three singing schemes with different input conditions: free singer, accompanied singer, and solo singer. We have also proposed a BEGAN based architecture that uses GRUs and grouped dilated convolutions to learn to generate singing voices in an adversarial way. For evaluating such models, we proposed several objective metrics and implemented a model to measure the compatibility between a given accompaniment track and the generated vocal track. The evaluation shows that the audio quality of the generated voices still leave much room for improvement, but in terms of humanness and emotion expression our models work fine.

Score and lyrics-free singing voice generation is a new task, and this work represents only a first step tackling it. How such models can contribute to computational creativity remains to be studied. From a technical point of view, there are also many interesting ideas to pursue. For example, we have chosen to extract only pitch-related information from the accompaniment track for building the accompanied singer, but a more interesting way is to let the model learns to extract relevant information on its own. In the near future, we plan to investigate advanced settings that allow for timbre and expression control, and experiment with other network architectures, such as coupling a fine-grained autoregressive model with a multiscale generation procedure as done in MelNet (Vasquez and Lewis 2019), or using multiple discriminators that evaluate the generated audio based on multi-frequency random windows as done in GAN-TTS (Bińkowski et al. 2019).

**References**


Automatic Dialect Adaptation in Finnish and its Effect on Perceived Creativity

Mika Hämäläinen\textsuperscript{1} 
mika.hamalainen@helsinki.fi

Niko Partanen\textsuperscript{2} 
niko.partanen@helsinki.fi

Khalid Alnajjar\textsuperscript{3} 
khalid.alnajjar@helsinki.fi

Jack Rueter\textsuperscript{1} 
jack.rueter@helsinki.fi

Thierry Poibeau\textsuperscript{4} 
thierry.poibeau@ens.fr

\textsuperscript{1}Digital Humanities, \textsuperscript{2}Finnish, Finno-Ugrian and Scandinavian Studies, \textsuperscript{3}Computer Science, University of Helsinki, FI
\textsuperscript{4}Lab. LATTICE, ENS/PSL & CNRS & Univ. Sorbonne nouvelle, FR

Abstract

We present a novel approach for adapting text written in standard Finnish to different dialects. We experiment with character level NMT models both by using a multidialectal and transfer learning approaches. The models are tested with over 20 different dialects. The results seem to favor transfer learning, although not strongly over the multi-dialectal approach. We study the influence dialectal adaptation has on perceived creativity of computer generated poetry. Our results suggest that the more the dialect deviates from the standard Finnish, the lower scores people tend to give on an existing evaluation metric. However, on a word association test, people associate creativity and originality more with dialect and fluency more with standard Finnish.

Introduction

We present a novel method for adapting text written in standard Finnish to different Finnish dialects. The models developed in this paper have been released in an open-source Python library\textsuperscript{1} to boost the limited Finnish NLP resources, and to encourage both replication of the current study and further research in this topic. In addition to the new methodological contribution, we use our models to test the effect they have on perceived creativity of poems authored by a computationally creative system.

Finnish language exhibits numerous differences between colloquial spoken regional varieties and the written standard. This situation is a result of a long historical development. Literary Finnish variety known as Modern Finnish developed into its current form in late 19th century, after which the changes have been mainly in the details (Häkkinen 1994, 16). Many of the changes have been lexical due to technical innovations and modernization of the society: orthographic spelling conventions have largely remained the same. Spoken Finnish, on the other hand, traditionally represents an areally divided dialect continuum, with several sharp boundaries, and many regions of gradual differentiation from one municipality to another municipality.

Especially in the later parts of 21th century the spoken varieties have been leveling away from very specific local dialects, and although regional varieties still exist, most of the local varieties have certainly became endangered. Similar processes of dialect convergence have been reported from different regions in Europe, although with substantial variation (Auer 2018). In the case of Finnish this has not, however, resulted in merging of the written and spoken standards, but the spoken Finnish has remained, to our day, very distinct from the written standard. In a late 1950s, a program was set up to document extant spoken dialects, with the goal of recording 30 hours of speech from each municipality. This work resulted in very large collections of dialectal recordings (Lyytikäinen 1984, 448-449). Many of these have been published, and some portion has also been manually normalized. Dataset used is described in more detail in Section Data and Preprocessing.

Finnish orthography is largely phonemic within the language variety used in that representation, although, as discussed above, the relationship to actual spoken Finnish is complicated. Phonemicity of the orthography is still a very important factor here, as the differences between different varieties are mainly displaying historically developed differences, and not orthographic particularities that would be essentially random from contemporary point of view. Thereby the differences between Finnish dialects, spoken Finnish and Standard Finnish are highly systematic and based to historical sound correspondences and sound changes, instead of more random adaptation of historical spelling conventions that would be typical for many languages.

Due to the phonemicity of the Finnish writing system, dialectal differences are also reflected in informal writing. People speaking a dialect oftentimes also write it as they would speak it when communicating with friends and family members. This is different from English in that, for example, although Australians and Americans pronounce the word today differently, they would still write the word in the same way. In Finnish, such a dialectal difference would result in a different written form as well.

We hypothesize that dialect increases the perceived value of computationally created artefacts. Dialectal text is something that people are not expecting from a machine as much as they would expect standard Finnish. The effect dialect has on results can be revealing of the shortcomings of evaluation methods used in the field.

\textsuperscript{1}https://github.com/mikahama/murre
**Related Work**

Text adaptation has received some research attention in the past. The task consists of adapting or transferring a text to a new form that follows a certain style or domain. As the particular task of dialect adaptation has not received a wide research interest, we dedicate this section in describing different text adaptation systems in a mode broad sense.

Adaptation of written language to a more spoken language style has previously been tackled as a lexical adaptation problem (Kaji and Kurohashi 2005). They use style and topic classification to gather data representing written and spoken language styles, thereafter, they learn the probabilities of lexemes occurring in both categories. Thus, we can learn the differences between the spoken and the written on a lexical level and use this information for style adaptation. The difference to our approach is that we approach the problem on a character level rather than lexical level. This makes it possible for our approach to deal with out-of-vocabulary words and to learn inflectional differences as well without additional modeling.

Poem translation has been tackled from the point of view of adaptation as well (Ghazvininejad, Choi, and Knight 2018). The authors train a neural model to translate French poetry into English while making the output adapt to specified rhythm and rhyme patterns. They use an FSA (finite-state acceptor) to enforce a desired rhythm and rhyme.

Back-translation is also a viable starting point for style adaptation (Prabhumoye et al. 2018). They propose a method consisting of two neural machine translation systems and style generators. They first translate the English input into French and then back again to English in the hope of reducing the characteristics of the initial style. A style specific bi-LSTM model is then used to adapt the back translated sentence to a given style based on gender, political orientation and sentiment.

A recent line of work within the paradigm of computational creativity presents a creative contextual style adaptation in video game dialog (Hämäläinen and Alnajjar 2019). They adapt video game dialog to better suit the state of the video game character. Their approach works in two steps: first, they use a machine translation model to paraphrase the syntax of the sentences in the dialog to increase the variety of the output. After this, they refill the new syntax with the words from the dialog and adapt some of the content words with a word embedding model to fit better the domain dictated by the player’s condition.

A recent style adaptation (Li et al. 2019) learns to separate stylistic information from content information, so that it can maximize the preservation of the content while adapting the text to a new style. They propose an encoder-decoder architecture for solving this task and evaluate it on two tasks: sentiment transfer and formality transfer.

Earlier work on Finnish dialect normalization to standard Finnish has shown that the relationship between spoken Finnish varieties and literary standard language can be modeled as a character level machine translation task (Partanen, Hämäläinen, and Alnajjar 2019).

**Data and Preprocessing**

We use a corpus called Samples of Spoken Finnish (Institute for the Languages of Finland 2014) for dialect adaptation. This corpus consists of over 51,000 hand annotated sentences of dialectal Finnish. These sentences have been normalized on a word level to standard Finnish. This provides us with an ideal parallel data set consisting of dialectal text and their standard Finnish counterparts.

The corpus was designed so that all main dialects and the transition varieties would be represented. The last dialect booklet in the series of 50 items was published in 2000, and the creation process was summarised there by Rekunen (2000). For each location there is one hour of transcribed text from two different speakers. Almost all speakers are born in the 19th century. Transcriptions are done in semi-narrow transcription that captures well the dialect specific particularities, without being phonetically unnecessarily narrow.

The digitally available version of the corpus has a manual normalization for 684,977 tokens. The entire normalized corpus was used in our experiments.

<table>
<thead>
<tr>
<th>Dialect</th>
<th>Short</th>
<th>Sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Etelä-Häme</td>
<td>EH</td>
<td>1860</td>
</tr>
<tr>
<td>Etelä-Karjala</td>
<td>EK</td>
<td>813</td>
</tr>
<tr>
<td>Etelä-Pohjanmaa</td>
<td>EP</td>
<td>2684</td>
</tr>
<tr>
<td>Etelä-Satakunta</td>
<td>ES</td>
<td>848</td>
</tr>
<tr>
<td>Etelä-Savo</td>
<td>ESA</td>
<td>1744</td>
</tr>
<tr>
<td>Eteläinen Keski-Suomi</td>
<td>EKS</td>
<td>2168</td>
</tr>
<tr>
<td>Inkerinsuomalaismurteet</td>
<td>IS</td>
<td>4035</td>
</tr>
<tr>
<td>Kaakkois-Häme</td>
<td>KH</td>
<td>8026</td>
</tr>
<tr>
<td>Kamuu</td>
<td>K</td>
<td>3995</td>
</tr>
<tr>
<td>Keski-Karjala</td>
<td>KK</td>
<td>1640</td>
</tr>
<tr>
<td>Keski-Pohjanmaa</td>
<td>KP</td>
<td>900</td>
</tr>
<tr>
<td>Länsi-Satakunta</td>
<td>LS</td>
<td>1288</td>
</tr>
<tr>
<td>Lansi-Uusimaa</td>
<td>LU</td>
<td>1171</td>
</tr>
<tr>
<td>Länspohja</td>
<td>LP</td>
<td>1026</td>
</tr>
<tr>
<td>Läntinen Keski-Suomi</td>
<td>LKS</td>
<td>857</td>
</tr>
<tr>
<td>Peräpohjola</td>
<td>P</td>
<td>1913</td>
</tr>
<tr>
<td>Pohjoinen Keski-Suomi</td>
<td>PKS</td>
<td>733</td>
</tr>
<tr>
<td>Pohjoinen Varsinais-Suomi</td>
<td>PV</td>
<td>3885</td>
</tr>
<tr>
<td>Pohjois-Häme</td>
<td>PH</td>
<td>859</td>
</tr>
<tr>
<td>Pohjois-Karjala</td>
<td>PK</td>
<td>4292</td>
</tr>
<tr>
<td>Pohjois-Pohjanmaa</td>
<td>PP</td>
<td>1801</td>
</tr>
<tr>
<td>Pohjois-Satakunta</td>
<td>PS</td>
<td>2371</td>
</tr>
<tr>
<td>Pohjois-Savo</td>
<td>PSA</td>
<td>2344</td>
</tr>
</tbody>
</table>

Table 1: Dialects and the number of sentences in each dialect in the corpus

Despite the attempts of the authors of the corpus to include all dialects, the dialects are not equally represented in the corpus. One reason for this is certainly the different sizes of the dialect areas, and the variation introduced by different speech rates of individual speakers. The difference in the number of sentences per dialect can be seen in Table 1. We do not consider this uneven distribution to be a problem, as it is mainly a feature of this dataset, but we have paid at-
tention to these differences in data splitting. In order to get proportionally even numbers of each dialect in the different data sets, we split the sentences of each dialect into training (70%), validation (15%) and testing (15%) the split is done after shuffling the data. The same split is used throughout this paper.

The dialectal data contains non-standard annotations that are meant to capture phonetic and prosodic features that are usually not represented in the writing. These include the use of the acute accent to represent stress, superscripted characters, IPA characters and others. We go through all characters in the dialectal sentences that do not occur in the normalizations, i.e. all characters that are not part of the Finnish alphabets and ordinary punctuation characters. We remove all annotations that mark prosodic features as these are not usually expressed in writing. This is done entirely manually as sometimes the annotations are additional characters that can be entirely removed and sometimes the annotations are added to vowels and consonants, in which case they form new Unicode characters and need to be replaced with their non-annotated counterparts.

**Automatic Dialect Adaptation**

In order to adapt text written in standard Finnish to dialects, we train several different models on the data set. As a character level sequence-to-sequence neural machine translation (NMT) approach has been proven successful in the past for the opposite problem of problem of dialectal or historical language variant to the standard language (see (Bollmann 2019; Hämäläinen et al. 2019; Veliz, De Clercq, and Hoste 2019; Hämäläinen and Hengchen 2019)), we approach the problem form a similar character based methodology. The advantage of character level models to word level models is their adaptability to out of vocabulary words; a requirement which needs to be satisfied for our experiments to be successful. In practice, this means splitting the words into characters separated by white-spaces and marking word boundaries with a special character, which is underscore (_) in our approach.

In NMT, language flags have been used in the past to train multi-lingual models (Johnson et al. 2017). The idea is that the model can benefit from the information in multiple languages when predicting the translation for a particular language a expression by a language specific flag given to the system. We train one model with all the dialect data, appending a dialect flag to the source side. The model will then learn to use the flag when adapting the standard Finnish text the the desired dialect.

Additionally, we train one model without any flags or dialectal cues. This model is trained to predict from standard Finnish to dialectal text (without any specification in terms of the dialect). This model serves two purposes, firstly if it performs poorly on individual dialects, it means that there is a considerable distance between each dialect so that a single model that adapts text to a generic dialect cannot sufficiently capture all of the dialects. Secondly, this model is used as a starting point for dialect specific transfer learning.

We use the generic model without flags for training dialect specific models. We do this by freezing the first layer of the encoder, as the encoder only sees standard Finnish, it does not require any further training. Then we train the dialect specific models from the generic model by continuing the training with only the training and validation data specific to a given dialect. We train each dialect specific model in the described transfer learning fashion for an additional 20,000 steps.

Our models are recurrent neural networks. The architecture consists of two encoding layers and two decoding layers and the general global attention model (Luong, Pham, and Manning 2015). We train the models by using the OpenNMT Python package (Klein et al. 2017) with otherwise the default settings. The model with flags and the generic model are trained for 100,000 steps. We train the models by providing chunks of three words at a time as opposed to training one word or whole sentence at a time, as a chunk of three words has been suggested to be more effective in a character-level text normalization task (Partanen, Hämäläinen, and Alnajjar 2019).

Table 2 shows an example of the sequences used for training. The model receiving the dialect flag has the name of the dialect appended to the beginning of the source data, where as the generic model has no additional information apart from the character sequences. The dialect specific transfer learning models are also trained without an additional flag, but rather the exposure solely to the dialect specific data is considered sufficient for the model to better learn the desired dialect.

**Results and Evaluation**

In this section, we present the results of the dialect adaptation models on different dialects. We use a commonly used metric called word error rate (WER) and compare the dialect adaptations of the test sets of each dialect to the gold standard. WER is calculated for each sentence by using the following formula:

\[
WER = \frac{S + D + I}{S + D + C}
\]  

(1)

WER is derived from Levenshtein edit distance (Levenshtein 1966) as a better measurement for calculating word-level errors. It takes into account the number of deletions \(D\), insertions \(I\) and the number of correct words \(C\).

The results are shown in Tables 3 and 4. On the vertical axis are the models. *Flags* represents the results of the model that was trained with initial tokens indicating the desired dialect the text should be adapted in. *No flags* is the model trained without any dialectal information, and the rest of the models are dialect specific transfer learning models trained on the *no flags* model.

The results are to be interpreted as the lower the better, i.e. the lower the WER, the closer the output is to the gold dialect data in a given dialect. These results indicate that the *no flag* model does not get the best results for any of the dialects, which is to be expected, as if it reached to good results, that would indicate that the dialects do not differ from each other. Interestingly, we can observe that the transfer
After this careful examination of the models, we proceed to the generation of dialectal poems and dialectology can reliably detect minute disfluencies in dialectal predictions, especially when the error is introduced by a dialect speaker or an advanced specialist in Finnish dialectology. There are also numerous examples of features that are in variation also within one dialect. In these cases the model has to be done for each dialect individually after training a generic model.

Evaluation of the models with and without dialectal flags shows that especially in word forms that are highly divergent in the dialect, it is almost impossible for the model to predict the correct result that is in the test set. This doesn't mean that the model's output would necessarily be entirely incorrect, as the result may still be perfectly valid dialectal representation, it just is in a different variety.

There are also numerous examples of features that are in variation also within one dialect. In these cases the model may produce a form different from that in the specific row of a test set. These kind of problems are particularly prominent in examples where the dialectal transcription contains prosodic phenomena at the word boundary level. Since the model starts the prediction from standard Finnish input, it cannot have any knowledge about specific prosodic features of the individual examples in test data. Some phonological features such as assimilation of nasals seem to be overgeneralized by the model, and also in this case it would be impossible for the model to predict the instances where such phenomena does not take place due to particularly careful pronunciation.

Another interesting feature of the model is that it seems to be able to generalize its predictions into unseen words, as long as they exhibit morphology common for the training data. There are, however, instances of clearly contemporary word types, such as recent international loans, that have general shape and phonotactics that are entirely absent from the training data. The problems caused by this are somewhat mitigated by fact that in many cases the standard Finnish word can be left intact, and it will pass within the dialectal text relatively well.

This has a consequence that the scores reported here are possibly slightly worse than the model's true abilities. The resulting dialectal text can still be very accurate and closely approximate the actual dialect, although the prediction would slightly differ from the test instances. At the same time if the model slips into predicted text some literary Finnish forms, the result is still perfectly understandable, and also in real use the dialects would rarely be used in entire isolation from the standard language.

It must also be taken into account that only either a native dialect speaker or an advanced specialist in Finnish dialectology can reliably detect minute disfluencies in dialectal predictions, especially when the error is introduced by a form of other dialect. Similarly it would be very uncommon to have such knowledge about all the Finnish dialects the model operates on. After this careful examination of the models, we proceed to the generation of dialectal poems and
their further evaluation by native Finnish speakers.

**Effect on Perceived Creativity**

In this section, we apply the dialect adaptation trained in the earlier sections to text written in standard Finnish. We are interested in seeing what the effect of the automatically adapted dialect is on computer generated text. We use an existing Finnish poem generator (Hämäläinen 2018) that produces standard Finnish (SF) text as it relies heavily on hand defined syntactic structures that are filled with lemmatized words that are inflected with a normative Finnish morphological generator by using a tool called Syntax Maker (Hämäläinen and Rueter 2018). We use this generator to generate 10 different poems.

The poems generated by the system are then adapted to dialects with the models we elaborated in this paper. As the number of different dialects is extensive and conducting human questionnaire with such a myriad of dialects is not feasible, we limit our study to three dialects. We pick Etelä-Karjala (EK) and Inkerinsuomalaismurteet (IS) dialects because they are the best performing ones in terms of WER and Pohjoisen Varsinais-Suomi (PVS) dialect as it is the worst performing in terms of WER. For this study, we use the dialect specific models tuned with transfer learning.

A qualitative look at the predictions revealed that the dialectal models have a tendency of over generating when a word chunk has less than three words. The models tend to predict one or two additional words in such cases, however, if the chunk contains three words, the models do not over nor under generate words. Fortunately this is easy to overcome by ensuring that only as many dialectal words are considered from the prediction as there were in the chunk written in standard Finnish. For instance *olen vanha* (I am old) gets predicted in IS as *olev vanha a*. The first two words are correctly adapted to the dialect, while the third word *a* is an invention by the model. However, the models do not systematically predict too many words as in *pieni ?* adaptation. For this reason, we only consider as many words as in the original chunks when doing the dialectal adaptation.

**Replicating the Poem Generator Evaluation**

In our first experiment, we replicate the poem generator evaluation that was used to evaluate the Finnish poem generator used in this experiment. We are interested in seeing whether dialectal adaptation has an effect on the evaluation results of the creative system. They evaluated their system based on the evaluation questions initially elaborated in a study on an earlier Finnish poem generator (Toivanen et al. 2012). The first evaluation question is a binary one *Is the text a poem?*. The rest of the evaluation questions are asked on a 5-point Likert scale:

1. How typical is the text as a poem?
2. How understandable is it?
3. How good is the language?
4. Does the text evoke mental images?
5. Does the text evoke emotions?
6. How much do you like the text?

The subjects are not told that they are to read poetry nor that they are reading fully computer generated and dialectally adapted text. We conduct dialectal adaptation to the 10 generated poems to the three different dialects, this means that there are altogether four variants of each poem, one in standard Finnish, and three in dialects. We produce the questionnaires automatically in such a fashion that each questionnaire has the 10 different poems shuffled in random order each time. The variants of each poem are picked randomly so that each questionnaire has randomly picked variant for each of the poems. Every questionnaire contains poems from all of the different variant types, but none of them contains the same poem more than once. Each questionnaire is unique in the order and combination of the variants. We
introduce all this randomness to reduce constant bias that might otherwise be present if the poem variants were always presented in the same order.

We print out the questionnaires and recruit people native in Finnish in the university campus. We recruit 20 people to evaluate the questionnaires each of which consisting of 10 poems. This means that each variant of a poem is evaluated by five different people.

Table 6 shows the results from this experiment, however some evaluators did not complete the task for all poems in their pile\(^2\). Interestingly, the results drop on all the parameters when the poems are adapted into the different dialects in question. The best performing dialect in the experiment was the Etelä-Karjala dialect, and the worst performing one was the Pohjoinen Varsinais-Suomi dialect all though it got the exact same average scores with Inkerinsuomalaismurteet on the last three questions. Now these results are not to be interpreted as that dialectal poems would always get worse results, as we only used a handful of dialects form the possibilities. However, the results indicate an interesting finding that something as superficial as a dialect can affect the results. It is to be noted that the dialectal adaptation only alters the words to be more dialectal, it does not substitute the words with new ones, nor does it alter their order.

In order to better understand why the dialects were ranked in this order, we compare the dialectal poems to the standard Finnish poems automatically by calculating WER. These WERs should not be understood as “error rates” since we are not comparing the dialects to a gold standard, but rather to the standard Finnish poems. The idea is that the higher the WER, the more they differ from the standard. Table 7 shows the results of this experiment. The results seem to be in line with the human evaluation results; the further away the dialect is from the standard Finnish, the lower it scores in the human evaluation. This is a potential indication of familiarity bias; people tend to prefer the more familiar language variety.

**Word Association Test**

In the second experiment, we are interested in seeing how people associate words when they are presented with a standard Finnish version and a dialectally adapted variant of the same poem. The two poems are presented on the same page, labeled as A and B. The order is randomized again, which means that both the order of poems in the questionnaire and whether the dialectal one is A or B is randomized. This is done again to reduce bias in the results that might be caused by always maintaining the same order. The concepts we study are the following:

- emotive
- original
- creative
- poem-like
- artificial
- fluent

The subjects are asked to associate each concept with A or B, one of which is the dialectal and the other the standard Finnish version of the same poem. We use the same dialects as before, but which dialect gets used is not controlled in this experiment. We divide each questionnaire of 10 poems into piles of two to reduce the work load on each annotator as each poem is presented in two different variant forms. This way, we recruit altogether 10 different people for this task, again native speakers from the university campus. Each poem with a dialectal variant gets annotated by five different people.

Table 8 shows the results of this experiment. Some of the people did not answer to all questions for some poems. This is reflected in the no answer column. The results indicate that the standard Finnish variant poems were considered considerably more fluent than the dialectal poems, and slightly more emotive and artificial. The dialectal poems were considered considerably more original and creative, and slightly more poem-like.

It is interesting that while dialectal poems can get clearly better results on some parameters on this experiment, they still scored lower on all the parameters in the first experiment. This potentially highlights a more general problem on evaluation in the field of computational creativity, as results are heavily dependent on the metric that happened to be chosen. The problems arising from this “ad hoc” evaluation practice are also discussed by (Lamb, Brown, and Clarke 2018).

<table>
<thead>
<tr>
<th>SF</th>
<th>EK</th>
<th>PVS</th>
<th>IS</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>himo on pala,</td>
<td>himo om palo,</td>
<td>himo om palo,</td>
<td>himo om paloo,</td>
<td>desire is a fire,</td>
</tr>
<tr>
<td>se sytty herkäst</td>
<td>se sytty herkast</td>
<td>se sytty herkast</td>
<td>se sytty herkast</td>
<td>it gets easily ignited</td>
</tr>
<tr>
<td>taas intona se kokoa</td>
<td>taas intona se kokova</td>
<td>taas inton se kokoko</td>
<td>taas inton se kokoha</td>
<td>again, as an ardor it shall rise</td>
</tr>
<tr>
<td>milloin into on elosa?</td>
<td>millo into on elosa?</td>
<td>millo innoo on elosa?</td>
<td>millo into on elosa?</td>
<td>when is ardor vivacious?</td>
</tr>
<tr>
<td>näämmekö, me,</td>
<td>näämmekö me,</td>
<td>näämmekö me,</td>
<td>näämmekö me,</td>
<td>we will see</td>
</tr>
<tr>
<td>ennen kuin into jää pois?</td>
<td>ennen ku into jää pois?</td>
<td>ennen ku into jää pois?</td>
<td>ennen ku into jää pois?</td>
<td>before ardor disappears?</td>
</tr>
<tr>
<td>mikäli innot pyysivät,</td>
<td>mikäli innot pyysivät,</td>
<td>mikäli innot pyysivät,</td>
<td>mikäli innot pyysivät,</td>
<td>if ardors stayed,</td>
</tr>
<tr>
<td>sini huomaist innon</td>
<td>sini huomaist inno</td>
<td>sini huomaist inno</td>
<td>sini huomaist inno</td>
<td>you would notice the ardor</td>
</tr>
<tr>
<td>minä alan maksamaa innon</td>
<td>mie alan maksamaa inno</td>
<td>maa ala maksamaa inno</td>
<td>mie ala maksamaa inno</td>
<td>I will start paying for the ardor</td>
</tr>
</tbody>
</table>

Table 5: An example poem generated in standard Finnish and its dialectal adaptations to three different dialects
Typical

Poem | Typical | Understandable | Language | Mental images | Emotions | Liking |
---|---|---|---|---|---|---|
SF | 87.2% | M | Mo | Me | M | Mo | Me | M | Mo | Me | M | Mo | Me | M | Mo | Me |
EK | 82.6% | 2.5 | 2 | 2 | 3 | 3 | 2.87 | 3 | 3 | 3.26 | 4 | 3 | 2.67 | 2 | 2 | 2.70 | 2 | 3 |
IS | 77.6% | 2.69 | 2 | 3 | 2.90 | 4 | 3 | 2.78 | 2 | 3 | 3.27 | 4 | 3 | 2.86 | 2 | 3 | 2.61 | 3 | 3 |
PVS | 77.0% | 2.51 | 2 | 2 | 2.80 | 2 | 3 | 2.58 | 2 | 3 | 3.27 | 4 | 3 | 2.86 | 2 | 3 | 2.61 | 3 | 3 |

Table 6: Results from the first human evaluation. Mean, mode and median are reported for the questions on Likert-scale.

Table 7: The distance of the dialectal poems form the original poem written in standard Finnish.

<table>
<thead>
<tr>
<th>SF</th>
<th>Dialect</th>
<th>No answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>emotive</td>
<td>48%</td>
<td>46%</td>
</tr>
<tr>
<td>original</td>
<td>40%</td>
<td>60%</td>
</tr>
<tr>
<td>creative</td>
<td>32%</td>
<td>64%</td>
</tr>
<tr>
<td>poem-like</td>
<td>46%</td>
<td>50%</td>
</tr>
<tr>
<td>artificial</td>
<td>50%</td>
<td>40%</td>
</tr>
<tr>
<td>fluent</td>
<td>74%</td>
<td>24%</td>
</tr>
</tbody>
</table>

Table 8: Results of the second experiment with human annotators.

Conclusions

We have presented our work on automatic dialect adaptation by using a character-level NMT approach. Based on our automatic evaluation, both the transfer learning method and a multi-dialectal model with flags can achieve the best results in different dialects. The transfer learning method, however, receives the highest scores on most of the dialects. Nevertheless, the difference in WERs of the two methods is generally small, therefore it is not possible to clearly recommend one over another to be used for different character-level data sets. If the decision is based on the computational power used, then the multi-dialectal model with flags should be used as it only needs to be trained once and it can handle all the dialects.

The dialect adaptation models elaborated in this paper have been made publicly available as an open-source Python library\(^1\). This not only makes the replication of the results easier but also makes it possible to apply these unique Finnish NLP tools on other related research or tasks outside of the academia as well.

Our study shows that automatic dialect adaptation has a clear impact to how different attributes of the text are perceived. In the first experiment that was based on existing evaluation questions, a negative impact was found as the scores dropped on all the metrics in comparison to the original standard Finnish poem. However, when inspecting the distance the dialects have from the standard Finnish, we noticed that the further away the dialect is form the standard, the lower it scores.

We believe that the low scores might be an indication of familiarity bias, which means that people have a tendency of preferring things they are more familiar with. Especially since the evaluation was conducted in a region in Finland with a high number of migrants from different parts of the country. This leads to a situation where the most familiar language variety for everyone regardless of their dialectal background is the standard Finnish variety. Also, as the dialectal data used in our model originates from the Finnish speakers born in the 19th century, it remains possible that the poems were transformed into a variety not entirely familiar to the individuals who participated into our survey. In the upcoming research it is necessary to investigate the perceptions of wider demographics, taking into account larger areal representation.

Based on our results, it is too early to generalize that familiarity bias is a problem in evaluation of computationally creative systems. However, it is an important aspect to take into consideration in the future research. We are interested in testing this particular bias out in the future in a more controlled fashion. Nevertheless, the fact that a variable, such as dialect that is never controlled in the computational creativity evaluations, has a clear effect on the evaluation results, raises a real question about the validity of such evaluation methods. As abstract questions on 5-point Likert scale are a commonly used evaluation methodology, the question of narrowing down the unexpected variables, such as dialect, that affect the evaluation results positively or negatively is vital for the progress in the field in terms of comparability of results from different systems.

Even though the initial hypothesis we had on dialects increasing the perceived value of computationally created artefacts was proven wrong by the first experiment, the second experiment showed that dialects can indeed have a positive effect on the results as well, in terms of perceived creativity and originality. This finding is also troublesome form the point of view of computational creativity evaluation in a larger context. Our dialect adaptation system is by no means designed to exhibit any creative behavior of its own, yet people are more prone to associating the concept creativity with dialectically adapted poetry.

The results of the first and second experiment give a very different picture of the impact dialect adaptation has on perceived creativity. This calls for a more thorough research on the effect different evaluation practices have on the results of a creative system. Is the difference in results fully attributable to subjectivity in the task, what was asked on how it was asked. Does making people pick between two (dialectal and standard Finnish in our case) introduce a bias not present when people rate the poems individually? It is

\(^1\)https://github.com/mikahama/murre
important these questions be systematically addressed in the future research.

Acknowledgments
Thierry Poibeau is partly supported by a PRAIRIE 3IA Institute fellowship ("Investissements d’avenir" program, reference ANR-19-P3IA-0001).

References

Hämäläinen, M., and Hengchen, S. 2019. From the past to the future: a fully automatic nmt and word embeddings method for ocr post-correction. In Recent Advances in Natural Language Processing, 432–437. INCOMA.


Shimon the Rapper: A Real-Time System for Human-Robot Interactive Rap Battles

Richard Savery, Lisa Zahray, Gil Weinberg
Georgia Tech Center for Music Technology
Atlanta, USA
rsavery3, lzahray3, gilw @gatech.edu

Abstract
We present a system for real-time lyrical improvisation between a human and a robot in the style of hip hop. Our system takes vocal input from a human rapper, analyzes the semantic meaning, and generates a response that is rapped back by a robot over a musical groove. Previous work with real-time interactive music systems has largely focused on instrumental output, and vocal interactions with robots have been explored, but not in a musical context. Our generative system includes custom methods for censorship, voice, rhythm, rhyming and a novel deep learning pipeline based on phoneme embeddings. The rap performances are accompanied by synchronized robotic gestures and mouth movements. Key technical challenges that were overcome in the system are developing rhymes, performing with low-latency and dataset censorship. We evaluated several aspects of the system through a survey of videos and sample text output. Analysis of comments showed that the overall perception of the system was positive. The model trained on our hip hop dataset was rated significantly higher than our metal dataset in coherence, rhyme quality, and enjoyment. Participants preferred outputs generated by a given input phrase over outputs generated from unknown keywords, indicating that the system successfully relates its output to its input.

Introduction
Interactive music systems have largely focused on generating instrumental music. Lyric generation and singing synthesis have been explored, but past research did not focus on vocal musical response to human input in real-time. The field of robotic musicianship uses embodiment to improve the relationship between humans and AI, inspiring humans in new creative ways. We combine these fields to create an interactive robotic system that improvises lyrically with a human in real-time. We select hip hop as a genre well-suited toward real-time improvisation, due to art forms like freestyle rapping and battle rap.

Shimon, seen in the left of Figure 1, is a marimba-playing robot who has recently been redesigned to have singing capabilities. Shimon collaborates with humans to write the lyrics to his own songs, taking keywords as input. However, in previous work, the lyric generation, voice synthesis, and gestures were not generated in real-time, and did not react to a human voice live on-stage. Our goal with this project was to allow Shimon to respond to a rapper in real-time with computer-generated rhyming lyrics, voice, rhythm, and gestures. We aim to provide the experience of a rap battle between a human and a robot, with the intention of inspiring the human rapper with machine-driven responses that are unlikely to be generated by humans.

This paper includes a technical overview of each sub-task of the system, beginning with analyzing voice input from a human rapper and ending with generating voice and synchronized gesture output by the robot. Throughout the paper, we discuss the key challenges faced during development and how they were overcome. Several examples of generated rap lyrics, rhyme analysis, and rhythm are provided. We also include a system evaluation using both quantitative and qualitative metrics. We analyze the overall perception of the system, various quality metrics of the output, and the system’s success at generating output to match its input. Videos examples of the system are available here 1. To our knowledge, this is the first system with a full working pipeline of vocal input from a rapper to vocal rap output in time to a beat.

Related Work
Generative and Interactive Music and Hip Hop
Computerized generative music systems have been widely explored from early systems in the 1950’s (Hiller 1968) to modern deep learning based systems (Briot, Hadjeres, and

1www.richardsavery.com/shimonraps
Pachet 2017), tending to focus on Western Classical music and Jazz. Stylistically closer to rap generation is Algorithmic Raving, where algorithms are used to create electronic dance music (Savery 2018), although its emphasis is instrumental music. Hip hop, however, has many unique stylistic features that distinguish it from other genres and present new challenges for generative systems. Linguistically hip hop is far extended from other poetic traditions or other music forms and uses a ‘highly intertextual’ form that ‘demonstrates multilayered poetic complexity’ (Alim 2003). Lyric delivery is commonly referred to as flow and - among many unique features - includes distinct approaches to meter, beat division, and rhyme placement (Condit-Schultz 2017; Komaniecki 2019).

Lyric Generation and Voice Synthesis

While computerized music generation has a long established history, lyric generation has only recently begun to receive attention. Past systems have focused on lyric generation based only on text without considering a musical melody, such as the Korean language lyric generator in (Son et al. 2019). Other efforts have focused on fitting lyrics to an existing melody in Tamil (Ramakrishnan A and Devi 2010), or for Jazz (Watanabe et al. 2018). Rap lyric generation has also been partly addressed in past work, although it has focused on hip hop as a standard natural language generation task (Karsdorp, Manjavacas, and Kestemont 2019), instead of a focus on hip hop’s unique aesthetic. These efforts also do not focus on real-time interaction, and often have a much more confined scope, such as generating text similar to 14 unique rappers (Potash, Romanov, and Rumshisky 2015).

State of the art results in singing synthesis have recently been achieved by deep learning based systems (Blauau and Bonada 2017), building on developments made by WaveNet(Oord et al. 2016). These systems, however, are computationally expensive to train and far too slow to be used for real-time generation. Comparative models that work in real-time using concatenation, such as Vocaloid (Kenncho and Ohshita 2007), have not matched results from offline models. Singing and voice synthesis have both been used extensively in robotic systems, such as the generation of robotic vocal prosody (Savery, Rose, and Weinberg 2019b; 2019a). To our knowledge, no vocal synthesis model has focused on rap. Additionally, we believe no system has attempted to both generate lyrics and synthesize the results in real-time for a robot to interact with a human performer.

System Overview

In the following section we present an overview of the system design of Shimon the Rapper. In particular we focus on three key challenges:
1. Latency and processing for real-time applications
2. Developing internal rhymes and rhythm
3. Approaches to censorship

The system is written in Python, with some interfaces to MaxMSP for easy audio processing and access to external plugins for processing on the generated voice. The standard

Audio Analysis

Audio from the rapper is recorded through a standard audio interface connected to an AT803 Omnidirectional Condenser Lavaliere. MaxMSP is used to record the audio, breaking the rappers incoming lyrics into smaller segments. The incoming audio is chunked with a simple volume threshold, with gaps in the lyrics separated when the length of time below the threshold is over 300 milliseconds. As each clip is chunked in real-time it is written to a wav file, and a UDP message is sent from MaxMSP to Python with the file name. Python then calls Google’s Speech to Text on each segment. While using an online speech-to-text system does add some latency, chunking means there is no latency cost above the last sample sent. By keeping chunks short, the added processing time averages around 1.5 seconds of latency. We also experimented widely with offline options, testing each API linked with Speech Recognition. However, we found Google Cloud Speech API to be the most reliable considering the background noise often present in live performances. We can opt to detect the end of the rapper’s phrase by either waiting for a silence of over 700 ms or a preset number of musical bars to pass. However, to decrease latency we often end the speech to text detection at random locations, allowing Shimon to start rapping, signifying to the human that their turn is over.

Text Analysis

Keywords are identified from the text once it has been extracted from the audio. We implemented the TextRank algorithm (Mihalcea and Tarau 2004) to categorize keywords. TextRank is a graph-based model used to rank the importance of text. In our implementation, text of up to 100 words is always processed in under 10 milliseconds. We then generate a list of synonyms and antonyms from wordnet (Oram 2001) for each keyword. Sentiment analysis is also implemented, as well as a system that categorizes which rapper

Text Rank

Figure 2: System Overview

interchange for the system involves a rapper free-styling over a loop for an undetermined amount of time followed by Shimon responding. The loop can be any musical material that has a set tempo.
from the dataset the incoming text is most similar to. We do not currently use these functionalities, as we have not found them helpful for the generation process.

**Dataset**

During lyric generation we primarily alternate between two custom-created datasets, switching between models in real-time. These data sets were created through a custom lyric web scraper: Verse Scraper ① This tool was created for two main reasons. Firstly, standard datasets group all lyrics for a song together, whereas hip hop commonly uses multiple rappers on the same track. We wanted to be able to associate lyrics with individual artists. The second benefit of our scraper is high level of customization in dataset creation, allowing us to create datasets from certain years, subsets of an artist catalogue, and other custom metrics. For our deep learning system we use either a hip hop dataset containing 25,000 songs or a metal dataset containing 15,000 songs. We were interested in comparing these two datasets to see how well metal music lyrics transfer to the hip hop genre.

**Phoneme Embedding and LSTM-RNN**

The primary novel element in our deep learning system is the use of a phoneme embedding layer. Phonemes are the second smallest layer of word vocalization, distinguishing how words are pronounced. Groups of phonemes create each syllable, and syllables create words. Phoneme embedding has been very rarely used in generative systems with one of the only uses being in a speech recognition system(Yenigalla et al. 2018) or occasionally in speech synthesis (Li et al. 2016). In purely text-based systems, preliminary work has shown that phoneme vector spaces contain distinctive feature contrasts to word embeddings (Silfverberg, Mao, and Hulden 2018). We contend that due to the unique linguistic properties of hip hop, phoneme embeddings offer a promising approach for a generative system. These properties are the unique relationship between words, built on a preference for rhyme from phonemes, over common semantic meaning. These rhymes also occur at any point in the lyrics, not only at the end of lines. Additionally, hip hop flow uniquely relies on phonemes (Edwards 2013) and often contains non-standard word variations and intentional variations in pronunciation to achieve flow.

The primary challenge of this approach was creating a dataset of phonemes. Mappings between the spelling of words and their phonemes are not always consistent. No extensive dataset currently exists of lyrics to phonemes, leading us to create a conversion process. While such conversion systems do exist, we found no system that can capture all the dialects used in hip hop. Our process begins by using CMU Pronouncing Dictionary② which is based on the ARPABET phonetic transcriptions. When words are not found in the dictionary, we attempt to break the word up into the most likely phoneme subsets by searching through the dictionary for subsets of phonemes that fit the word. After phoneme subsets are found for the word, that word is then added to the dictionary so that all repeats of a word are treated the same.

With phonemes as the embedding layer, we can use a relatively standard deep learning model. While state of the art models such as GPT-2(Radford et al. 2019) are based on Transformer with attention layers, we used an encoder/decoder RNN-LSTM, as we aimed for real-time generation. Many of the advantages of larger models are for improved long term structure, which is not required for short phrases such as the ones we are creating.

We first generate many lines of text with the model. We then automatically choose phrases from all the generations that utilize either a keyword, or a synonym or antonym from the keywords. This process allows us to combine multiple generations and meanings, while still placing an emphasis on internal rhymes created through deep learning, and line by line rhymes through rhyme detection.

**Censorship**

Censorship of the output was a significant consideration and design challenge for the system. We first created a list of 28 words that would not be appropriate for the system to output. For some words this list included multiple spelling variations. After creation, the list was encoded with ROT13, to allow us to more comfortably share the code.

We considered multiple approaches to censorship, aiming to balance maintaining authenticity of the dataset, while meeting language requirements. In original tests we considered excluding from the hip hop dataset any song that contained one of the words in our list of censored words. This reduced the data size from 25,000 songs, to 7,000. To counter this we considered removing lines or whole verses containing certain words. Given hip hop’s reliance on flow, this approach proved ineffective as it seemed to significantly alter the data set. Likewise we considered replacing offending words with a substitute, but again, due to subtle elements impacting rhythm and flow this was deemed as an inappropriate method. To maintain authenticity we decided to keep the original dataset and instead censor phrases by post processing created material. After creation we discard any generation that includes a filtered word and create a new generation as a replacement. While this does add extra processing time into the system, we found it a worthwhile trade off.

**Rhyme Detection and Choices**

The phoneme embedding naturally generates lines of text containing internal rhymes. We automatically select which generated lines to use based on the quantity and quality of internal rhymes of each line, as well as which lines rhyme best with each other. We originally tried using existing rhyme libraries, but found that they were too slow when iterating through a large number of words. We created our own implementation for scoring rhymes that runs in a few hundredths of a second on large numbers of phrases.

We are interested in detecting and scoring two types of rhyme: perfect rhymes, where vowels and consonants match, and slant rhymes, where words have similar but not identical sounds. For our system, we specify slant rhymes as vowels that match, but consonants that may not match. We

①https://github.com/RFirstman/versescraper
②http://www.speech.cs.cmu.edu/cgi-bin/cmudict

Proceedings of the 11th International Conference on Computational Creativity (ICCC’20)
ISBN: 978-989-54160-2-8
create dictionaries for each line, recording phoneme patterns and their frequency of occurrence within the line. Our first dictionary type is for perfect rhymes, which records the last one, two, and three-syllable sequences (vowels and subsequent consonants) of each word. The second dictionary is for slant rhymes, which records the last two and three-vowel sequences of each word. Finally, we create a dictionary excluding words that are only 3 or fewer phonemes long. This allows us to give a lower score to words that rhyme perfectly but are very short (such as "to" and "do"). For all dictionaries, a sequence of vowels or syllables is only added if it contains at least one stressed vowel.

We first use these dictionaries to select the line with the highest internal rhyme score. Each type of phoneme pattern is assigned a score, where perfect rhymes and a higher number of matching syllables are scored higher than slant rhymes and fewer matching syllables. These scores are summed according to the number of instances of each detected rhyme. We do not count multiple occurrences of the same word as any type of rhyme.

To select each subsequent line, we find the line that rhymes best with the previously selected line. We do this by calculating the rhyme score for phoneme patterns that are present in both lines. After all lines are selected, we assign each rhyme group a unique number to identify which words rhyme with each other. This information is used in the next steps of the pipeline. Figure 3 shows examples of generated lines that were selected using the rhyme scoring, along with the calculated rhyme scores.

**Text to Rhythm**

We next generate rhythms for the generated text, with each syllable assigned a time. We designed a rule based system for rhythmic generation based on concepts presented by Edwards (Edwards 2009; 2013). Edwards compiled a collection of interviews with over one-hundred leading hip hop artists discussing their approach for flow and rhythm. Based on these interviews, we designed a rhythm generation that is able to map to any tempo, although it has been primarily used for tempos ranging from 80 to 160 beats per minute.

Rhythmic generation focuses on emphasizing rhyming words. Emphasis is added to words by expanding the length of the word beyond non-rhymes and by placing different lengths of silence after each rhyming word. Keywords with more than one syllable are set as quarter-note triplets, allowing them to stand out from non-keywords without interrupting the flow. All non-rhyming words are set as eighth notes.

In experiments we also applied similar rules to nouns or other word types, but found this was not represented in texts and was not well received internally. Figure 4 shows an example of a generated phrase with its corresponding rhythm.

**Rhythm to Voice**

Shimon’s voice is generated by modifying the output of Google’s text-to-speech system. Speech Synthesis Markup Language (SSML) provides options for changing vocal prosody in text-to-speech systems. We use SSML to emphasize any word that rhymes with at least one other word. We additionally pitch-shift all matching rhymes by the same amount, allowing for either upwards or downwards pitch shifts. This method, which is common in hip hop (Komaniecki 2019) can help the listener to notice which words rhyme with each other.

In order to place the words at the correct time according to the generated rhythm, we originally tried running text-to-speech separately on each word. However, we found that this took too much time and could result in an unnatural cadence when the words were strung together. Therefore, in order to quickly generate the audio, we run text-to-speech once on the entire sentence with added breaks between each word using SSML. We then split the resulting audio file into the non-silent segments to separate the words.

We found that the endings of individual words are frequently cut off upon generation, due to the way words flow together in natural conversation. This made it more difficult to generate arbitrary rhythms from the words, as a long break after a cut-off word could sound odd and difficult to understand. To address this issue, as well as to better match the generated rhythm for multi-syllable words, we time-stretch the words to flow more naturally into each other.

We align each synthesized word to start at the time given by the rhythm generation. We end each word’s audio on whichever occurs first: the start time of the following word, or a tempo-dependent offset after the start time of the word’s last syllable. This helps words that are close together flow into each other more naturally, while also stretching words.
that precede longer gaps to mitigate any cut-off endings.

While our generated rhythm provides start times for each syllable in a word, we only modify word start and end times when generating the audio. A more complex audio analysis could have allowed for alignment of each syllable. However, we chose not to do this to increase system robustness, and to maintain the original timings produced by the text-to-speech system to preserve naturalness. Finally, we compress and filter the output to improve the audio quality, using commercial audio plugins\(^6\). This also raises the overall pitch of Shimon’s voice, producing a unique and cute voice timbre befitting of Shimon’s persona as a robotic rapper.

**Gesture Synchronization**

Shimon’s gesture design while rapping consist of both synchronizing his mouth movements to the audio, as well as generating head and neck movements throughout the rap battle. We create the mouth movements using each syllable’s times from our rhythm generation. Similarly to the key pose approach used in (Tachibana, Nakaoka, and Kenmochi 2010), Shimon’s lip syncing linearly interpolates between phoneme-dependent positions. Some examples of these positions can be seen in Figure 5. The linear interpolation allows for smooth easing into and out of mouth poses. Consonants are given a default maximum duration. However, if a syllable’s vowel duration is shorter than the default consonant duration, half of the syllable time is given to the vowel and the remaining time is evenly divided among consonants. If a word is not found in the CMU Pronouncing dictionary, it is assigned the phonemes [P, AH, P] by default.

We raise Shimon’s eyebrows on rhyming words to increase emphasis. We also do this with the intention of conveying that Shimon is pleased with his generated rhymes and is challenging the human rapper interacting with him. When Shimon is listening to the rapper, we have him nod to the beat, move his head side to side on every downbeat, and move his body up and down on every other downbeat. We position his head and body so he is looking at the rapper. While performing his own rap, Shimon slowly moves side to side and up and down with the beat, keeping his head positioned so that his mouth is visible.

**Latency**

Latency is a constant trade-off between time and quality. The most time-intensive tasks are rap-to-text, generating phrases to select from, and rhythm-to-voice. All other tasks

are on the order of hundredths of a second or lower. Figure 6 shows the time required for each sub-process, highlighting the time difference between different numbers of generated words. We find that generating around 3,000 words is a good compromise for maintaining high quality with low latency. This number can be higher for a more powerful GPU. With these settings, the time between starting the generation and when Shimon begins rapping is approximately 11.69 seconds. However, because we can start the generation while the human rapper is still finishing, the perceived latency can be made to be lower. By ignoring the rapper’s last sentence, that time can be reduced to 6.69 seconds, during which an instrumentalist can play a quick solo or Shimon can use gestures to stall while generation completes.

The pipeline makes use of two computers, one that uses a 1080 GPU to generate choices for output phrases, and another that performs all other audio and computational tasks. Two computers are used due to compatibility with MaxMSP.

**System Evaluation**

A broad Turing style test, as is often used (Agres, Forth, and Wiggins 2016), does not make sense for this system since by definition, Shimon lyrical output does not aim to sound like a human. Likewise, computer based NLG or chatbot metrics tend to focus on features that are not easily applied to our design - such as readability and grammatical correctness - and have been shown to give significantly different results than human ratings for creative tasks (Novikova et al. 2017). There are multiple non-academic frameworks that exist to evaluate human-created hip hop and rap, however many of these tools were referenced in the creation process and would be unfairly biased towards our system \(^7\).

Additionally, throughout the design and development stage we regularly engaged with Atlanta rapper Dashill Smith. This involved five extended sessions where he informally analyzed and reviewed the system output. These sessions led to an iterative design process, where we would build on and alter the system based on his reactions and in-

---

\(^6\)https://polyversemusic.com/products/manipulator/

\(^7\)https://www.rappad.co/blueprints/faq
sights. During this stage of the project, we chose not to collect formal data from Smith, instead allowing for natural discussion and broad ideas for future directions and improvements. While evaluation by experts through interaction has been shown as an effective means to analyze interactive systems (Bown 2015), we chose to frame our final evaluation around audiences’ perception, enjoyment, and rating of the system, since while the system is interaction-based its ultimate use case is in musical performance to listeners. In future research we aim to engage multiple rappers with the system for evaluation.

With these challenges in mind we designed our evaluation to answer the following research questions:

1. What is the perception of the system by listeners and what do subjects think about idea of a robot-human having rap battles in general?
2. Can we create high quality stand alone hip hop?
   2.1. Do our stand alone rap outputs lead to good coherence, rhythm, rhyme, quality, and enjoyment?
   2.2. Are there differences in these metrics when using the hip hop dataset versus the metal dataset?
3. Is there a clear relationship between the system’s output and its input?

Method

33 participants answered survey questions about videos and text samples generated by the system. The participants were undergraduate students recruited from the Georgia Tech School of Psychology participant pool. Participants were not required to have any musical experience, however chose to participate in the experiment based on their interest in the topic. We calculated the minimum amount of time it should take participants to complete the survey, watching all videos and reading all text samples, and eliminated 6 participants who completed it in less than that amount of time. This left us with 27 remaining participants.

First, participants were introduced to the project’s concept by watching a 45-second video clip of a rapper freestyle rapping back and forth with Shimon. They were asked to describe their thoughts on the footage. This data was used to answer Research Question 1. In past studies analyzing text has been shown to provide insightful information on robot perception (Vlachos and Tan 2018). In generating each sample shown to participants for the remainder of the survey, we ran the model three times on its keyword input and hand-selected one response we believed to be the highest quality.

To address Research Question 2, we presented subjects with 10 randomly-ordered videos of Shimon performing a rap with subtitles, without being shown the input to the system. To generate the raps for these videos, 5 distinct sets of keywords were used. Each keyword set generated two of the samples, one using the hip-hop dataset and the other using the metal dataset. The participants were asked to rate the coherence, rhythm, rhyme, overall quality, and overall enjoyment of each sample on a scale from 1 to 7.

To address Research Question 3, participants were given 10 randomly-ordered tasks selecting which of two text samples they preferred as a response to given input text. One of these responses was generated by the model in response to the given input, and the other response was generated by a keyword unrelated to the input text. All keywords were randomly sampled from words that occurred over 10 times in the dataset. The order in which the responses were presented was randomized as well. Within each question, the two responses were generated using the same dataset, where 5 questions used the hip-hop dataset and 5 used the metal dataset.

Results

Perception From the collected text responses we firstly analyzed the sentiment of each response, using the Valence Aware Dictionary and sEntiment Reasoner (VADER) (Hutto and Gilbert 2014). This provided us with a value for negative, positive, neutral and compound sentiment (see Fig.7). The compound sentiment is a normalized, weighted composite score between -1.0 (negative) and 1.0 (positive). The mean of the compound sentiment was 0.33, indicating an overall positive perception of the system.

Subjects’ comments covered a wide range from ‘very expressive’ and ‘amazing’ to ‘the robot’s voice sounded very strange when juxtaposed with the human’. A common thread was participants describing the generation as ‘better than expected’. We also found most comments focused on the voice with less emphasis on the lyrics. See Figure 8 for a word map with the most common words in subjects responses (excluding standard stopwords and the word robot).

Rap Quality and Data Set Comparison In each of the categories, the hip hop data set achieved a slightly higher mean (see Fig 9). Comparing the hip hop and metal dataset
using an independent samples t-test we found two insignificant results for rhythm (p = 0.226) and quality (p=0.225). This makes sense as rhythm is generated independently of the data set and quality should consider the system as a whole. The enjoyment was significant with hip hop being slightly favored (p = 0.046). We also found significant results in the coherence(p=0.027) and rhymes (p=0.017).

We found the lowest correlation between rhythm and enjoyment, while there was a strong correlation between the perceived quality and coherence, rhythm, and rhymes (see Fig 10). Importantly, we found no clear correlation between any category and the participants actual rated enjoyment of the rap, perhaps implying we need to consider other metrics for our generation system.

Input and Output  To address Research Question 3, we evaluated whether participants preferred lyrics generated from the given input over lyrics generated from unknown, random keywords. We assigned each participant a score, defined as the number of times (out of the 10 questions) they preferred the lyrics generated by the given input over an unknown random input. We then performed a 1-tailed, 1-sample t-test on these scores, comparing against an expected mean of 5 out of 10. The p-value is 0.00033, which is less than the alpha of 0.05. The average score across all participants was 6.2 out of 10. The data support that participants preferred responses generated from the given input over samples generated from randomized keywords. This supports that participants recognized that the system’s output related to its input. However, the number of compared samples was small, so it is possible there were other reasons for this preference.

Each task within the system has room for improvement in quality and latency. Occasionally, our rhyme detection system may miss or incorrectly identify rhymes if the word is not found in the CMU Pronouncing dictionary, or if it has multiple possible pronunciations. More work into word pronunciation given context in a sentence would help improve both the phoneme embedding and rhyme scoring tasks.

Future work in rhythm generation could use a data-based approach, as opposed to our strictly rule-based system. MCFlow (Condit-Schultz 2017) is one example of a dataset that could be useful for this. It would also be interesting to approach rhythm generation for different styles of rap.

Currently, the head and body gestures are predetermined, with only eyebrow and mouth gestures dependent on the rap’s content. Incorporating computer vision could allow for more personalized interactions, such as following the rapper as they move across the room, and potentially matching the way they move to the beat.

The pipeline we have established allows for modifying the settings and even the overall approach to each subtask. As we gain more feedback from rappers interacting with the system, we will continue making improvements. We hope to use this system to inspire rappers through the novel experience of an interactive rap dialogue with a robot.
Acknowledgements

We would like to thank Eddy Chiao, Brian Model, Jacob Meyers, Naveen Ram, Rob Firstman, and Spencer Gold for their contributions to prototyping the system.

References


Being Creative: A Cross-Domain Mapping Network

Jichen Wu, Maarten H. Lamers, Wojtek J. Kowalczyk
Leiden Institute of Advanced Computer Science
Leiden University
Leiden
j.wu.13@umail.leidenuniv.nl, m.h.lamers@liacs.leidenuniv.nl, w.j.kowalczyk@liacs.leidenuniv.nl

Abstract
The ability to extract features from objects or concepts and to connect them in a meaningful way is believed to be crucial to creativity. This paper proposes a novel computational model for creative behaviour, by learning extractions and connections separately. Such separation enables adaptive feature and object connections, which means one object can be connected to different other objects within different contexts. This paper applies recent cognitive theories of computational creativity to two specific tasks: an image-to-image mapping and music-to-video mapping. In both cases deep neural networks are used to automate these two creative cognition tasks.

Introduction
It is not easy to answer the question: what is creativity? Nevertheless, many agree that the ability to make connections or combinations between concepts is a key component of creativity. Mekern, Hommel, and Sjoerds (2019) implied a unitary model of creative cognition. This model, which is inspired by Hommel and Wiers (2017), is based on adaptive relations between features of concepts and ideas. “Features of concepts” means that the relations are not directly made between concepts but through features, and that one concept is represented by many independent features. “Adaptive” implies the weighting of possible features according to different situations so that the contributions of different features vary to adapt to the current situation. This unitary creative model is arguably an integration and simplification of the divergent creativity model (Kenett et al. 2018) and the convergent creativity model (Kajić et al. 2017).

On the practical side, however, few computational models that stress features and their relations were built into machines. Some previous studies did partly model these properties: Olteteanu and Falomir (2016) used an “Object Replacement and Composition” system, in which the features of an object seem to have to be manually decided and the size of the dataset is relatively small. This approach makes the dataset accurate and human-understandable. However, as they pointed out, the system needs a larger dataset or a different approach to build the feature space to perform on a larger scope. In another creative machine (Augello et al. 2016), features of an image object are extracted directly based on color and texture information. Although this approach avoided manually building datasets, it took the risk of losing much information, for example, distribution of color, or shapes within the image. These issues reveal the need for a more general and automatic method to efficiently acquire more comprehensive features and relations.

Luckily, the development of machine learning makes it possible to extract various kinds of useful features from large datasets. Therefore, given two domains $X$ and $Y$, we can extract features from points in these domains into feature spaces $Z_X$ and $Z_Y$ and consider the problem of finding various mapping functions that map $Z_X$ onto $Z_Y$ that satisfy some context dependent constraints. We should notice here that, as we explain in Related Work, the term ‘feature’ has a somewhat different meaning in the context of ‘machine learning’ and ‘cognition research’. Nevertheless, without any risk of confusion we will use this term without specifying the context.

Previous studies (Augello et al. 2016; Olteteanu and Falomir 2016; Huang et al. 2018; Liu, Breuel, and Kautz 2017) can only find a single mapping function because this function relies on some level of equality between the feature vectors. For example, a red, thick T-shirt should be mapped to red, thick trousers. However, humans do not always relate things by similarity of features. For example, we might think a red, thin T-shirt makes a good pair with blue, thick trousers because of the influence of fashion trends or because they are similarly rare in their domains. To find multiple mapping functions we propose two criteria. First, if humans experience two things together for several times, they may naturally connect them. So if previously experienced pairs from $X$ and $Y$ are provided, a mapping function should map their feature vectors together. This criterion is named: previously experienced mapping. Second, similar objects in domain $X$ should be mapped to similar objects in domain $Y$, while dissimilar objects in $X$ should be mapped to dissimilar objects in $Y$. This criterion is named: topology mapping.

While a few previous models (Huang et al. 2018; Zhu et al. 2017b) can already learn non-deterministic mappings between domains, they still cannot learn different mappings based on different criteria. Rather the mapped data points are randomly selected based on statistic distributions. Our research serves as a realization for recent creative cognition models as well as an exploration of creativity in contemporary deep learning models. It aims to connect cognition...
theory and computational applications.

In the next section, related work regarding the use of features in computational creativity theories and its relevance in computational technologies is introduced. The methodology and framework of our Cross-Domain Mapping Network (CDMN) are described in Method, followed by experiments assessing the effectiveness and creative behaviours of the CDMN. Finally, conclusions are provided.

Related Work

Related work is discussed in four parts. First, the trend of considering features in creative cognition is introduced. Secondly, machine learning methods for feature encoding are briefly described. Third, methods for constructing feature connections are recognized from the field of machine learning, specifically image-to-image mapping networks. Finally, possibilities for multi-modal translations (audio, image, video) are provided.

Features and Computational Creativity

A feature set is a distributed representation of a concept, where ‘distributed’ means a feature set is maximally representative of the concept on a certain level of significance. Feature extraction and feature connections have an important role in the theory of computational creativity since the distinction between convergent and divergent thinking of creativity (Guilford 1967). Divergent thinking is to generate creative ideas or solving problems creatively by exploring many possible solutions whereas convergent thinking is to provide a single best solution to a well-defined question. The ability to make associations is believed to be important to both processes (Gabora 2010; Mednick 1962).

Both divergent and convergent thinking are modeled explicitly in a dual-process computational painter by Augello et al. (2016). The key operation in this painter is to replace image part A with image B that shares features such as color and texture. However, in such dual-process models, outcomes of divergent and convergent processes are often hard to distinguish and outcomes of each process rely to some extent on the interplay between these two processes (Mekern, Hommel, and Sjoerds 2019).

Merken et al. (2019) proposed a unitary model for creative behaviour in which creative behaviours are facilitated by the interplay between features. To achieve a degree of flexibility, under a different context different connections between features are activated. From Merken’s unitary model, we identify two processes: (1) the encoding of distributed features, and (2) flexible connections between features to facilitate contextualization and individual differences. Along with these two processes, three criteria are proposed to evaluate a creative model: (1) features are distributed and representative, (2) the connections are flexible under different circumstances, and (3) individual differences (flexible and persistent) can be modeled.

Encoding and Decoding Features

Although the importance of features is identified, the computational creativity community still does not have many reliable and automatic ways to encode (extract) features from concepts. Previous programs either involve manual encoding (e.g. (Olteanu and Falomir 2016)) or the encoded features are not well-distributed (a concept cannot be fully represented by its feature set; e.g. (Augello et al. 2016)). Fortunately, neural networks have been developed to find complex features from raw data. A prominent example of such networks is the autoencoder (e.g. (Hinton and Salakhutdinov 2006)). One may think that the features encoded by an autoencoder are just numbers and thus of a different nature from ‘color’, ‘shape’ or ‘texture’. However, what makes features crucial is not whether they are abstract numbers, activation of neurons or concrete attributes, but whether they serve as a set of distributed representations of a concept or an object. The encoding-decoding process of an autoencoder ensures that the encoded features are maximally representative of the input data.

One property of an autoencoder is that the encoded features are data-specific. With a dataset of cup images, it can never recognize ‘concave’ as a feature because it is not distinctive. Because of this, creative problem solving as by Olteanu and Falomir (2016) cannot be easily achieved without an extensive dataset. However, this property also enables autoencoders to generate never-before-seen objects from features. Utilizing this advantage, this paper aims at a generative model instead of an associative model.

Feature Connection

Several recent studies regarding Domain Transfer Networks (DTNs) have already achieved the process of feature encoding and connection to some extent. However, relevance between DTNs and creativity theories is hardly mentioned. Thus, in this section, the relevance of DTNs in the connecting of features is identified.

DTNs have been studied for image-to-image translation. Early works (Yoo et al. 2016; Zhu et al. 2017a) do not perform any manipulations on the encoded features. Some successors (Zhu et al. 2017b; Taigman, Polyak, and Wolf 2017) found that having features as inputs to the generators enhanced the performance of the network. However, in these methods, features from different domains are not explicitly connected.

Liu et al. (2017) were the first to model explicitly the connection of features, although this connection simply equates features of X with features of Y without flexibility of such connections. Huang et al. (2018) proposed an improvement that provides some degree of flexibility of connections. It implies that some features are always connected and other features are never connected. Since true flexibility is deciding which features are relevant to the context and should be connected adaptively, our work makes an attempt to implement such flexibility.

Cross-modal Temporal Data Generation

We are interested not only in the domain of images. The problem becomes more tricky if the domains include temporal data, such as video or music, or require cross-modal (audio to visual, visual to audio) mapping. With other applications showing the possibilities (Song et al. 2019;
Ephrat and Peleg 2017), we want to explore and evaluate the creative behaviour in audio-video translation also.

**Method**

The two steps of creative cognition (feature encoding and flexible feature connecting) are implemented in a computer program. Although the programs are specially designed for image data and temporal data, the methodology should in principle be suitable for other types of data as well.

**Mapping Functions**

A mapping function \( m \) is, in principle, a function that translates each data point \( z_x \) in feature space \( Z_X \) to a data point \( z_y \to y \) in feature space \( Z_Y \): \( z_x \to y = m(z_x) \). Since usually there are infinitely many points in a feature space, the number of mapping functions is infinite. To attack this problem we first cluster \( Z_X \) and \( Z_Y \) into \( n_X \) and \( n_Y \) clusters and construct mapping functions to map clusters upon each other. Thus a mapping function from \( Z_X \) to \( Z_Y \) exists among the finite set of possibilities with size \((n_Y)^{n_X}\).

However much information could be lost if the feature vector only carries clustering information. To overcome this, a feature vector \( z_x \) is split into two vectors: a cluster vector \( c_x \) in a finite space and a vector carrying other detail information \( v_x \) in an infinite space. Next, the feature extraction function \( E_X \) is defined: \( c_x, v_x = E_X(x) \). Similarly we have \( E_Y \). While there are \((n_Y)^{n_X}\) possible mappings from the cluster vector space of \( X \) to the cluster vector space of \( Y \), \( v_x \) is passed unchanged: \( v_x \to y = v_x \). The relation between \( c_x \) and \( v_x \) can be understood using Fig. 1. While \( c_x \) defines the center of a cluster in the space, \( v_x \) is a small vector deviating from this center. With clusters numbers \( n_X = n_Y = 1 \), this model is identical to the shared latent space assumption proposed by Liu et al. (2017). With \( n_X \to \infty, n_Y \to \infty \) and \( v \to 0 \), in theory it is possible to construct any arbitrary mapping function. The process of mapping a data point \( x \) to domain \( Y \):

\[
\begin{align*}
    c_x, v_x & = E_X(x) \\
    c_x \to y & = m(c_x) \\
    y & = G_Y(c_x, v_x)
\end{align*}
\]

where \( G_Y \) is a function that decodes a feature vector back into the domain \( Y \), or \( y = G_Y(y) \) where \( y \approx y \).

![Figure 1: Relations between c and v and how a mapping function works.](image)

To find good mapping functions we apply two criteria mentioned in the Introduction: previously experienced mapping and topology mapping. First, when previously experienced pairs \( \{(x_1, y_1), \ldots, (x_{mn}, y_{mn})\} \) are present, assuming that \( c_{x_i}, v_{z_i} = E_X(x_i) \), the loss of a mapping function can be measured by how well it matches clusters of given pairs:

\[
L_{\text{pair}} = \sum_{i=1}^{mn} w_i \cdot \text{eval}(m(c_{x_i}), c_{y_i})
\]

where \( \text{eval}(m(c_{x_i}), c_{y_i}) \) returns 0 if \( m(c_{x_i}) = c_{y_i} \) otherwize, and \( w_i \) is a weight assigned to each pair \( i \).

Second, when topology mapping is used, we first assume that the topology is preserved with the feature extraction functions \( E_X \) and \( E_Y \) so similar objects have small distance in feature space. The next loss function measures how well a mapping function \( m \) preserves the distances between clusters of \( x \):

\[
L_{\text{topo}} = \sum_{i=1}^{n_X} \sum_{j=1}^{n_X} (d(c_i^x, c_i^y) - d(m(c_i^x), m(c_i^y)))^2
\]

where \( c_i^x \) represents the \( i \)-th cluster of the total \( n_X \) clusters and \( d(c_i^y, c_j^y) \in [0, 1] \) is the normalized Euclidean distance between \( c_i^x \) and \( c_j^y \). \( L_{\text{topo}} \) is called the stress function in multidimensional scaling.

Given weights \( w_{\text{topo}} \), the overall loss to be minimized is:

\[
L_{\text{map}} = L_{\text{pair}} + w_{\text{topo}} L_{\text{topo}}
\]

An algorithm to find good solutions of \( L_{\text{map}} \) is subject to the criteria that the mapping functions should be able to model individual differences. In this paper genetic algorithms are used because their design can facilitate the modeling of flexible and persistent (explorative and exploitative) individuals.

**Network for Feature Encoding**

The functions \( E_X, G_X, E_Y, G_Y \) are learned by a neural network. We assume that \( c_x, v_x, c_y, v_y \) are of the same dimensionality. Furthermore, \( c_x, c_y \) can be represented by one-hot vectors \( h_x, h_y \):

\[
\begin{align*}
    c_x & = H_X(h_x), \quad c_y = H_Y(h_y)
\end{align*}
\]

We update \( E_X \) so that: \( h_x, v_x = E_X(x) \) and update \( m \) so that \( h_x \to y = m(h_x) \) (which does not change the functionality of \( E_X \) and \( m \)). The same change is also made to \( E_Y \). Makhzani et al. (2015) have shown that this way it is possible for the encoder to learn cluster representations via one-hot vectors.

The complete structure of the network is shown in Figure 2. It has two functions. When passing \( c_x, v_x \) to \( G_X \), it is an autoencoder to reconstruct \( x \), when passing \( H_Y(m(h_x)), v_x \) to \( G_Y \), it is a mapping network. For networks \( E \) and \( G \), similar structures to Liu et al. (2017) are used.

**Training for autoencoding**

The two autoencoder structures are trained independently. For autoencoder \( (E_X, H_X, G_X) \), we want reconstructions \( \hat{x} \) to approach inputs \( x \). Here L1 loss is used:

\[
L_{\text{recon}}^x = E[||x - \hat{x}||]
\]

where \( \hat{x} = G_X(H_X(h_x), v_x) \) and \( h_x, v_x = E_X(x) \). \( v_x \) and \( v_y \) need to follow the same distribution for the mapping to work. We let them both follow the standard distribution.
$N(0, I)$ where $I$ is the identity matrix. This is done by using a VAE structure (Kingma and Welling 2014) that uses KLDivergence loss:
\[ L_{KL}^v = KL(E(X|x)[1]|N(0, I)) \]
where $E(X|x)[1] = v_x$ and $h_x$ is expected to be a one-hot vector representing unsupervised clustering information. Adversarial training is used with a discriminator $D_X$ that tries to tell $h_x$ from a random real one-hot vector $h_x^r$ of the same dimensionality, resulting in loss function:
\[ L_{adv}^v = E[log(1 - D_X(E(X|x)[0]))] + E[log(D_X(h_x))] \]
where $E(X|x)[0] = h_x$. An illustration of the training process is shown in Figure 3.

**Figure 3:** Network structure at training for autoencoding loss.

Similarly, we have loss functions for $y$, and a total loss:
\[ L_{total}^x = w_{recon} \cdot (L_{recon}^x + L_{KL}^x) + w_{adve} \cdot (L_{adv}^x + L_{adv}^y) \]
which is minimized by $E$, $H$ and $G$ while maximized by $D$.

**Training for mapping** There are two problems if the network is only trained minimizing $L_{total}^x$. First, even though $L_{KL}$ penalizes $v_x, v_y$ that do not follow the Gaussian distribution, they nonetheless tend to deviate (Makhzani et al. 2015), especially in high dimensional spaces. This leads to the situation that $v_x$ and $v_y$ follow different distributions and the mapping $G_y(c_y, v_x)$ will only generate a noisy output. Second, the clustering has strong bias — there could be one cluster containing half of the training set while most other clusters are empty. To overcome these issues, a joint training process is designed. With $v_x$ and a random vector $h_y^r$, $G_y(H_y(h_y^r), v_x)$ learns to generate a realistic image with the help of a discriminator $D_y^{img}$ for generated outputs:
\[ L_{y}^{GAN} = E[1 - D_y^{img}(G_y(H_y(h_y^r), E_X(x)[1])]] + E[log(D_y^{img}(y))] \]
This ensures that $G_y$ learns the distribution of $v_x$ and also the full distribution $h_y$. Cycle consistence loss (Zhu et al. 2017a) is also used:
\[ L_{cyc}^v = E[||v_x, h_y^r - E_y(G_y(H_y(h_y^r) + v_x))||_1] \]
where $v_x = E_X(x)[1]$. These processes are illustrated in Figure 4.

**Figure 4:** Network structure and training for mapping loss.

Similarly, we have the loss for $y \rightarrow x$ and a total loss of:
\[ L_{total}^y = w_{GAN} \cdot (L_{GAN}^x + L_{y}^{GAN}) + w_{cyc} \cdot (L_{cyc}^x + L_{cyc}^y) \]
which is minimized by $E$, $H$ and $G$ while being maximized by $D$. During the training process, the network is trained on $L_{total}^x$ and $L_{total}^y$ iteratively.

**Network for cross-modal temporal data** Besides unimodal image-to-image translation tasks, a creative model should also be able to solve cross-modal non-static (temporal) translation tasks. Such tasks post more restrictions on the autoencoder networks. A variant of the network shown in Figure 2 is specifically designed for audio-to-video translation with two generators consisting of LSTM convolutional blocks (Xingjian et al. 2015). The new functions are named $G^m_X$ and $G^m_Y$. $\hat{x}, mem_x^{t+1} = G^m_X(c_x, v_x, mem_x^t)$ where $mem_x^t$ is the memory of the network $G^m_X$ at time $t$.

**Experiments**
This section is divided into two parts. First, configurations and technical properties of the model are studied. Then, scenarios are provided to study the model’s creative behaviour.
**Performance Analysis**

This section studies configurations and technical properties of (1) an image-to-image mapping model, and (2) a music-to-video mapping model. The results are mostly qualitative as quantitative results can hardly be captured and evaluated.

**Image-to-image mapping**

The evaluation is performed using a dataset of clothing images that were obtained from the Alibaba Tianchi Big Data Competition\(^1\) entitled ‘Key-points Detection of Apparel – Challenge the Baseline’ and converted to 64x64 pixels. Images of blouses ($n \approx 5000$) are used for domain $X$, trousers and skirts ($n \approx 15000$) are used for domain $Y$.

First, the network’s sensitivity to hyperparameter settings is evaluated. Testing for distributions of $v$ (either $N(0, I)$ or $N(0, 0.1I)$), decoder normalization (none or layer normalization, cf. Meyer, Pfaffl, and Ulbrich (2010)), $w_{adv}$, $w_{rec}$, $w_{cyc}$, and $w_{GAN}$. Figure 5 shows the results of five hyperparameter sets. Most sets were able to capture some distinctive features of each top clothing item but the relations with bottom clothing items are difficult to observe. There does not seem to be a heavy reliance on hyperparameters and we arbitrarily selected set 3 ($v \sim N(0.1I)$; no decoder normalization, $w_{adv}$, $w_{rec}$, $w_{cyc}$, $w_{GAN} = 10, 1, 1, 1$) for the remainder of this section.

![Figure 5: Comparison of five hyperparameter sets. Each generated image is produced by $G_x$ from the corresponding $v_y$ and $h_x$.](image)

Next, our model is tested on known image-to-image translation datasets. These datasets are defined by that there is only one mapping rule that is human-interpretable. For example, in the ‘edges2shoes’ dataset (Zhu et al. 2017a) it would only make sense to pair a shoe image with an edge image if the contour of the shoe image is the same as the edge image.

In Figure 6, the results of application to the ‘edges2shoes’ dataset are shown. The network is first trained to minimize $L_{adv}^{total}$ and $L_{map}^{total}$. Then, a single mapping function $m_{deter}$ is learned from previous pairs. Figure 6 shows generated shoe images that have similar contours as the corresponding edge images although they do not match equally well as other state-of-the-art networks (Liu, Breuel, and Kautz 2017). This might indicate that the network is unable to encode enough information into vector $v$.

![Figure 6: Edges to shoes dataset. Mapping functions based on previous pairs.](image)

Next, the network is tested on a novel task: generating top clothing to match bottom clothing. In this scenario multiple mapping rules are needed, because different people, in different situations, have different rules of how clothing should be paired. Here two rules are tested: (1) topology mapping, and (2) color matching. The goal is to evaluate our model’s adaptivity to different rules. Adaptation to topology mapping is expected to generate similar items of top clothing when given similar images of bottom clothing. Adaptation to color matching is expected to generate top clothing that has the same color as the provided bottom clothing. Previous pairs for color matching are top and bottom items with similar color.

In Figure 7, topology mapping CDMN output is compared to that of UNIT (Liu, Breuel, and Kautz 2017) and color matching CDMN output is compared to that of Pix2pix (Isola et al. 2017). We see that while UNIT tends to generate tops that correspond to a given bottom (opposite color, similar texture, and shape), topology mapping CDMN finds a structural relation in which similar inputs result in similar outputs without correspondence of color, texture, and shape. Color matching CDMN can reproduce tops matching several major colors, although Pix2pix is more accurate in color matching. However, it is important to note that when the mapping rule changes, Pix2pix must learn from the beginning, while CDMN needs to find a new mapping function only.

![Figure 7: Comparison between outputs from different networks/ mapping functions.](image)

**Music visualization**

In this section, the model’s performance on music-to-video mapping is assessed. Although we focus on video generation from music, but not vice versa,
since the network is trained bidirectionally even unidirectional generation partially illustrates the model’s bidirectional behaviour.

The music dataset used is an arbitrary selection from the Million Song Dataset (Bertin-Mahieux et al. 2011) of 1000 songs that cover 10 genres and are reduced to 30 seconds duration each. The video dataset contains 10 videos of 200 frames each showing generated ‘jumping’ circles, as illustrated in Figure 8.

Figure 8: Ten successive frames of one video of the jumping circles dataset.

Figure 9 shows results from our model\(^2\) with four hyperparameter sets, differing in music window sizes, cluster counts \(n_1, n_2\), and numbers of LSTM convolutional blocks in the decoder. Due to limited space we do not show the detailed configurations of the four sets, but they are not essential for later observations. We observe that Set 1 generates video that appears inconsistent and lacks an observable pattern corresponding to the music input signal. Set 2 generates a green blob in each frame that seems to expand with music signal amplitude increases. Set 5 appears to result in similar behaviour, but with very subtle frame differences. Using hyperparameter Set 7, the blue element at center-left of the frames appears to shrink in anticipation of rising music signal amplitudes, whereas it expands after the signal peak passes.

Figure 9: Illustration of video output for four different hyperparameter sets, and the music input signal. Each image in a row represents a frame with 40 ms interval (25 Hz).

By comparing these four hyperparameter sets, it becomes apparent that this model is difficult to tune for good video output; ‘good’ in the sense that viewers observe patterns in the video that temporally correspond with the music. This does not imply that the model cannot uncover patterns in the input music, but it could result in patterns within output videos that are too subtle to observe. Defining a proper criterion to ensure that changes in video are not too subtle nor too dramatic remains an open problem.

---

\(^2\)Music visualization videos at vimeo.com/368386488

Creativity Evaluation

The next important question is: how creative is this model? We do not intend a thorough evaluation of creativity using external criteria in this paper. Instead we provide information and impression of the creative behaviour of our model from a technical point of view. We use the criteria mentioned earlier: (1) features are distributed and representative; (2) connections between domain spaces are flexible under different circumstances; and (3) flexible and persistent individual can be modeled. The first criterion is inherently met through the application of autoencoders. We define two tasks to assess our model’s agreement with criteria (2) and (3).

The first task focuses on the flexibility of mappings, specifically our model’s behaviour in the face of environmental and intentional changes. Imagine a scenario where a fashion designer has her own style based on many years of experience. Recently she attended a fashion show, which subconsciously changed her style preferences. Then she met a new friend who is a famous fashion designer. Impressed by his talent she wants to mimic his style. Next, she found that her style was too similar to other designers, in particular to that of her friend, and she decides to create her own unique style. However, her old experiences and style are rooted and not easy to change.

Can our model mimic such human-like behaviour? To align this scenario with our artificial designer, let’s assume that the designer’s job is that, when provided with a clothing bottom (trousers or skirt), she must create a top. Her experience can be represented by the many pairs of tops and bottoms that she has seen (Guide Set 1, Figure 10), while the fashion show exposure is Guide Set 2, and her friend’s style is Guide Set 3. Tops from all guide sets are randomly selected to match the given bottoms. Items in the set Creation 1 are generated solely based on Guide Set 1. Items in set Creation 2 are generated from Guide Sets 1 and 2 with random weights for all pairs of Set 1 in \([0, .9, 1.1]\) and for Set 2 in \([0.8, 1]\). Items in set Creation 3 are generated from Guide Sets 1 and 2 with the same weights as previous and random weights for all pairs from Guide Set 3 in \([2, 3]\). Finally, set Creation 4 is generated with random weights for Set 1 in \([0.9, 1.1]\), for Set 2 in \([-0.2, 1]\), and for Set 3 in \([-3, -1]\). Topology mapping with a small weight is also added for each creation so clusters that are not covered by the guide sets can be mapped.

Results are shown in Figure 10. These include creations based on never-before-seen bottoms. We observe that Creation 1 matches Guide Set 1 well. Inclusion of Guide Set 2, as expected, only directly changes a few creations (e.g. columns 5 and 8). However, this small impact may also affect future creations. For example, in column 10 Guide Set 2 contains a pink-red top. After learning from Guide Set 3 which has no impact on column 10, the artificial designer creates a red top. Creation 3 matches with Guide Set 3 well. It is interesting to see that Creation 4 is similar but not identical to Creation 1. This implies that the artificial designer does not immediately disregard old experience (Guide Set 1) in the face of new experiences, but that its style is shifted slightly. The experiment shows that the
The model can create flexible mapping functions under changing circumstances. These flexible behaviours simulate human creativity on some levels.

![Figure 10: Guide sets and creations for flexibility evaluation. Each image of a top in a guide set is paired with the bottom in the same column. Each created image is based on the input bottom of the same column.](image)

Our second task evaluates the modeling of persistent and flexible human individuals. While a persistent individual aims to find a single best solution, a flexible individual tends to explore for more possibilities, perhaps not all equally good. This difference can be modeled through the design choices of an evolutionary algorithm (EA). A persistent individual is modeled by a greedy EA with \((1 + \lambda)\) selection where \(\lambda\) are all offspring closest to the parent. A flexible individual is modeled by an EA with \((15, 30)\) selection. Each (persistent or flexible) individual takes its previous individual as one solution in its first EA generation and then runs the EA to minimize the loss of topology mapping. Here we do not have solid quantitative evaluation of music-visualization as it is very complex, if possible. Instead we evaluate it perceptually and qualitatively. We show that this already provides valuable information. Results are shown in Figure 11. We observe that persistent individuals 2 and 3 cannot create something new beyond persistent individual 1. Contrastingly, flexible individuals 2 and 3 are not restricted by prior experience encoded in flexible individual 1. However, judging by the loss, later generations of flexible individuals do not necessarily improve on earlier generations. Furthermore, even though persistent individuals have a lower loss, visually it is hard to say if the persistent individuals find ‘better’ mapping functions than the flexible individuals\(^3\). In fact, it is hard to assess whether or not an elaborated mapping function is better than a completely random mapping function!

Such difficulty implies that the model fails in carrying through relevant information from the music input to the visual output of the model. Perceptual consistency in the mapping appears missing. For music frames \(x_1, x_2, x_3\), and mapping \(x_1 \rightarrow y_1, x_2 \rightarrow y_2\) and \(x_3 \rightarrow y_3\), if a listener finds that \(\text{difference}(x_1, x_2) < \text{difference}(x_1, x_3)\), one would expect that visually \(y_1\) appears more similar to \(y_2\) than to \(y_3\). This is however not perceived in the output of our model.

Besides potential optimization issues for the network, this also reveals the more profound problem that the nature of finite cluster space and topology mapping make it hard to find a ‘good’ mapping. Clustering discretizes and originally infinite space, forcing areas of the original space to ‘disappear’ from consideration. One-dimensional topology mapping takes only Euclidean distance between clusters into account, implying that multi-dimensional relations between clusters are simplified and limiting the range of potentially generated output. In terms of the domains, would it make sense that a cluster of red jackets is closer to red T-shirts than to blue jackets? However, at this point, finite clustering with topology mapping is the only method for mapping. Future work may want to improve from here.

**Conclusion**

This work proposes the cross-domain mapping network (CDMN), a method for adaptive mappings and cross-domain content generation. It encodes finite cluster features and infinite individual variation thereupon from one domain, maps these cluster features to cluster features of a second domain, and from there generates (decodes) instances within the second domain. Different from previous work, the separation between encoding-decoding functions and mapping functions is modeled more towards replication of human creative behaviour and enables the mapping functions to adapt to changing situations. The use of mapping criteria based on topological distances within both domains and previous pairs helps the CDMN to show some complex human-like behaviours, as demonstrated in our scenario-based experiments.

We made an attempt to bridge creative cognition theory and machine learning applications. On one hand, as a GAN application this model achieves a higher level of creativity in terms of better adaptivity and individuality, when compared to prior work. Furthermore, as a realization of computational creativity theories, our model provides a highly automated method with which the unitary action control model with feature distribution and connection is shown to be computable. It shows the possibilities of using machine learning tools as a convenient and powerful method to build creative models and evaluate theories about creative cognition and
psychology.

Possible future work is suggested in two directions. In the direction of computational modelling, the limitations brought by finite clustering and topology mapping should be addressed. It is also possible to construct continuous feature spaces and design mapping functions conditional on regionality within those feature spaces, as opposed to applying indexed clustering in feature spaces. Moreover, it is not trivial to tune hyperparameters as more in-detail analysis can be performed with more detailed models. In the direction of cognition theories, mapping rules used in this paper can arguably model somewhat but limited human-like behaviors. This is because the mapping rules proposed in this paper are ad-hoc, which is due to the fact that how mapping functions are controlled is not well-known in cognitive psychology (Hommel and Wiers 2017). This paper addresses the importance of such studies to achieve closer-to-human level creativity in computational models.

References


WeirdAnalogyMatic: Experimenting with Analogy for Lyrics Transformation

Hugo Gonçalo Oliveira
CISUC, Department of Informatics Engineering
University of Coimbra, Portugal
hroliv@dei.uc.pt

Abstract

This paper is on the transformation of text relying mostly on a common analogy vector operation, computed in a distributional model, i.e., static word embeddings. Given a theme, original song lyrics have their words replaced by new ones, related to the new theme as the original words are to the original theme. As this is not enough for producing good lyrics, towards more coherent and singable text, constraints are gradually applied to the replacements. Human opinions confirmed that more balanced lyrics are obtained when there is a one-to-one mapping between original words and their replacement; only content words are replaced; and the replacement has the same part-of-speech and rhythm as the original word.

Introduction

Materialisations of human creativity rarely start from scratch. Consciously or not, artists are inspired by what they experience, including other artists and their creations. This is also true in the scope of Computational Creativity, where many systems rely on an inspiration set. When it comes to linguistic creativity, poetry generation systems may rely on a corpus of human-created poems where templates (Toivainen et al. 2012; Colton, Goodwin, and Veale 2012) or language models (Potash, Romanov, and Rumshisky 2015; Yan 2016) are acquired from; the initial population of a genetic algorithm is derived from (Hamalainen and Alnajjar 2015), both known for keeping syntactic and semantic regularities. Analogies are solved with the following operation on the vectors of the involved words \( \vec{a} - \vec{d} + \vec{b} \approx \vec{b} \) (e.g., a common example is king – man + woman ≈ queen).

Our main goal is thus to explore to what extent we can rely on word embeddings for transforming the semantics of a poem, in such a way that its theme shifts according to the seed, while text remains syntactically and semantically coherent. Transforming text, rather than generating it from scratch, should help to maintain the latter. For this, we make a rough approximation that the song title summarises its theme and every word in the lyrics is related to this theme. Relying on this assumption and recalling how analogies can be computed, shifting the theme is a matter of computing analogies of the kind ‘what is to the new theme as the original title is to a word used?’.

However, we soon noticed that text resulting from the exclusive application of the analogy operation had a series of issues. Therefore, we describe some constraints introduced towards better lyrics, e.g., to guarantee that functional words are not changed, syntax is coherent, or the original metre is kept. Yet, although more constraints lead to better structure and singability, they lower the chance of selecting related words, with a negative impact on the theme shift. To analyse the impact of different constraints on aspects like grammar, semantics, novelty or singability, a selection of results with different constraints was subjected to a human evaluation, which suggested that there should be a one-to-one mapping between original words and their replacement, only content words should be replaced, and the replacement must have the same part-of-speech and rhythm as the original word.

Although our experiments were performed in song lyrics, this would work similarly in any kind of poetry, or other textual genres.

The proposed approach constitutes the engine of a system for lyrics transformation, which we baptised as WeirdAnalogyMatic (WAM) because the obtained results could potentially be followed in the creation of parodies from known songs, popularised by artists such as Weird Al Yankovic – e.g., with hits like Eat it (transformation of Michael Jackson’s Beat it), Smells Like Nirvana (based on Nirvana’s Smells Like Teen Spirit), or Like a Surgeon (based on Madonna’s Like a Virgin). This kind of parody has also featured several comedy TV shows (e.g., Saturday Night Live) and advertising campaigns (e.g., These Bites are Made for Poppin’ a 2006 Pizza Hut ad by Jessica Simpson for Super Bowl, which is a transformation of These Boots are Made for Walkin’; or the 2000 TV ad for Mountain Dew, a trans-
formation of Queen’s Bohemian Rhapsody). All of those examples suggest that attempting at the automation of this creation procedure may be worth.

The remainder of the paper briefly overviews different approaches for poetry and song lyrics generation, with a focus on those that, along the way, exploit word embeddings. We then describe our approach and illustrate with the result of adding more constraints, step-by-step. Before concluding, we present the results of the evaluation survey, together with examples of the most and least appreciated lyrics.

Related Work

Poetry generation has long been a research topic in Computational Creativity, with much work during the last 20 years (Gonçalo Oliveira 2017). A prominent approach is the generation based on templates, instantiated by similes (Colton, Goodwin, and Veale 2012), instances of other relations (Gonçalo Oliveira and Oliveira Alves 2016), or by replacing certain words with others, with the same part-of-speech (PoS) (Agirrezabal et al. 2013), or associated to a target subject (Toivanen et al. 2012). While templates generally guarantee that syntactic rules are met, towards semantic coherence, poetry generators often have to rely on a model of semantics. For this, semantically-related words can be acquired from semantic networks (Agirrezabal et al. 2013; Gonçalo Oliveira and Oliveira Alves 2016), models of word associations (Toivanen et al. 2012), or of distributional semantics, such as word embeddings (Ghazvininejad et al. 2016; Hämäläinen and Alnajjar 2019a).

Alternative approaches to text generation, including creative text, are based on language models, which can be learned from large corpora with recurrent neural networks (Yan 2016), often with LSTM layer(s) (Potomanov, and Rumshisky 2015). Yet, recently, the generation of different kinds of text has been attempted with larger transformer-based language models, like GPT-2 (Radford et al. 2019), fine-tuned for a specific domain. In any of the previous, the first step is to learn word embeddings from a corpus on the target domain.

Not so different from template-based, one last alternative for producing new text is to start with a single original text and replace some of its words towards the desired intent. Such an approach was used for generating lyrics for parodies inspired by daily news (Gatti et al. 2017), achieved by expanding content words of a headline with WordNet (Fellbaum 1998) and Wikidata, then used for replacing words in original lyrics, having in mind syntactic (PoS) and metric constraints. Distributional semantics was not considered. Another common application is in the generation of shorter texts, like headlines (van Stegeren and Theune 2019) or slogans (Repar et al. 2018), where domain vocabulary can be expanded with word embeddings.

Also in the context of creative systems, the operations of similarity, neighbours, theme, and analogy in a word embedding space were formalised and used for producing song lyrics, with the selection of replacement words constrained by the given intentions (e.g., form, theme, sentiment) (Bay, Bodily, and Ventura 2017). Out of those operations, we focus exclusively on analogy and assume that all words in the lyrics are somehow related to a theme, which we approximate by the song title. The paper is focused on experiments and necessary workarounds for taking advantage of analogy and still have a result that is not only syntactically and semantically coherent, but also singable.

Step-by-Step Approach

Our goal is to transform a given text, so that it is still meaningful, but its semantics shifts to a new theme $t_n$, given by a single word. For this, we assume that every word $w_o$ in the original text is somehow related to a fixed meaning in a distributional space, seen as the original theme $t_o$. We then rely on analogy for computing new words $w_n$ for replacing each $w_o$. In our experiments, we use song lyrics and make the rough approximation that $t_o$ can be obtained from the song title, i.e., we use a model of distributional semantics for computing $t_o$ as the weighted average of the vector of all content words in the title. Since, at least in the tested models, words are ordered according to their frequency in the training corpus, we used their index in the model as their weight. This can be seen as a cheap approximation to word relevance, because more frequent words (i.e., less relevant) will have a lower index, thus lower weight, while less frequent ones (i.e., more relevant) will have a higher index.

To wrap it up, we assume that every word $w_o$ in the original lyrics is to the theme $t_o$, as a new word $w_n$ is to a new theme $t_n$. So, once a new theme $t_n$ is selected, we can, for every $w_o$, apply the 3CosAdd analogy solving method to the vectors of the involved words, and compute $w_n$ as follows:

$$w_n = w_o - t_o + t_n.$$ 

Yet, we soon realised that following this with no additional constraints resulted in text that was both hard to sing and ungrammatical. For minimising those issues, some constraints were added to the process of lyrics transformation. Such constraints are thoroughly described in this section, with their impact illustrated by results obtained. Different models of word embeddings were tested, but all results reported were obtained with the Stanford GloVe word vectors\(^2\) (Pennington, Socher, and Manning 2014), with 300 dimensions, pre-trained in a corpus of 6B tokens from Wikipedia and Gigaword 5. Though originally applied to word2vec models, 3CosAdd is also applicable to GloVe and generally achieves better performance in semantic analogies (Pennington, Socher, and Manning 2014).

Examples presented here used as input the lyrics of Smells Like Teen Spirit (hereafter, SLTS), by Nirvana\(^3\), with original lyrics in figure 1. The original theme is given by $t_o = \alpha.smells + \beta.like + \gamma.teen + \delta.spirit$, where $\alpha, \beta, \gamma$, and $\delta$ are the index-based weights. In this case, $t_o$ is a vector close to smells, the word with higher index value.

\(^1\)Even if this does not hold for many lyrics, it was our experimentation setting. Alternative theme approximations may consider the average embedding of the first line, the chorus, or the full song.

\(^2\)https://nlp.stanford.edu/projects/glove/

\(^3\)Given the controversy around the actual meaning of these lyrics and their connection with the title, this was, arguably, not the best choice, but it suits the purpose of illustrating the procedure.
Load up on guns, bring your friends
It’s fun to lose and to pretend
She’s over-bored and self-assured
Oh no, I know a dirty word
Hello, hello, hello, how low
Hello, hello, hello, how low
Hello, hello, hello
With the lights out, it’s less dangerous
Here we are now, entertain us
I feel stupid and contagious
Here we are now, entertain us
A mulatto, an albino, a mosquito, my libido

Figure 1: Original lyrics of Smells Like Teen Spirit.

**Only analogy**

The first attempt to test our hypothesis was to rely exclusively on analogy. Each line of the lyrics was first tokenized with the Stanford CoreNLP toolkit\(^4\) (Manning et al. 2014), and then every single word \(w_o\) in the original lyrics was replaced by a new word \(w_n\), that would be to the new theme \(t_n\) as \(w_o\) was to the original theme \(t_o\). Results immediately confirmed that this would not be enough for our purpose. This is illustrated in figure 2, where the title and first two lines of SLTS are presented for two different values of \(t_n\).

<table>
<thead>
<tr>
<th>(t_n)=computational,</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Title:</strong> computation mathematical mathematical theoretical</td>
</tr>
<tr>
<td>computation theoretical theoretical theoretical,</td>
</tr>
<tr>
<td>theoretical mathematical</td>
</tr>
<tr>
<td>theoretical’s mathematical mathematical mathematical</td>
</tr>
<tr>
<td>mathematical mathematical mathematical</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(t_n)=art, <strong>Title:</strong> arts works arts arts</th>
</tr>
</thead>
<tbody>
<tr>
<td>loads work arts gun, work work arts</td>
</tr>
<tr>
<td>part’s arts work losing arts work do</td>
</tr>
</tbody>
</table>

Figure 2: Lyrics for STLT with only analogy.

**Keep stopwords**

Although we could expect that computed analogous words would have the same PoS, thus ensuring that the grammatical structure was kept, this was not always the case. A possible cause is that, because functional words are used in many contexts, they end up having a limited contribution to the topic. This is illustrated in figure 2, where the title and first two lines of SLTS become:

<table>
<thead>
<tr>
<th>(t_n)=art, <strong>Title:</strong> arts works arts arts</th>
</tr>
</thead>
<tbody>
<tr>
<td>loads work arts gun, work work arts</td>
</tr>
<tr>
<td>part’s arts work losing arts work do</td>
</tr>
</tbody>
</table>

To avoid repetition and guarantee a one-to-one correspondence between \(w_o\) and \(w_n\), a map can be used for the history of replacements made. Such that, when selecting a word \(w_n\) for replacing \(w_o\), the history may be looked up and, if \(w_n\) was already used as a replacement for a different word than \(w_o\), it is not used. Instead, out of the words not previously used in the lyrics, the most similar to the analogy is selected. This is tested for each word in the model, ranked according to its similarity to the computed analogy vector, until a usable word is found or a predefined rank is reached. For the reported experiments, the maximum rank was 2,500, meaning that, at most, the 2,500 most similar words were tested.

We stress that, even if lower-ranked words will probably not be exactly an analogy, they should be similar to \(t_n\), or on the same topic. Once the map is integrated, with the previous values for \(t_n\), the first five lines of the lyrics of SLTS become those in figure 4.

**Part-of-Speech tagging**

Controlling repetition lead to better lyrics, but also made still existent grammatical issues more clear. Examples include losing, in the second line of both lyrics shown, or lines like physics no, i understand a methods translation or gallery no, i what a painting literature. Therefore, to select words \(w_n\) that match the PoS of a target \(w_o\), we did the following: (i) PoS-tag each line in the original lyrics with the CoreNLP PoS tagger; (ii) For each open \(w_o\) (nouns, verbs, adjectives, adverbs) and candidate replacement \(w_n\), create a new line where \(w_o\) is replaced by \(w_n\); (iii) PoS-tag the new line; (iv) If the sequence of PoS tags is the same as the original, use \(w_n\).

---

\(^4\)https://stanfordnlp.github.io/CoreNLP/
\(^5\)https://github.com/stanfordnlp/CoreNLP/blob/master/data/edu/stanford/nlp/patterns/surface/stopwords.txt
**t_n=computational**

**Title:** computational mathematical theoretical creativity

computation up on rifles, solve your colleagues
it's mathematics to losing and to algorithms
she's over-bored and probabilistic
physics no, i understand a methods translation
bioinformatics, bioinformatics, bioinformatics, how molecular

**t_n=art**

**Title:** art works arts museums

loads up on gun, work your friend
it's exhibition to losing and to do
she's over-bored and exhibited
gallery no, i what a painting literature
photography, photography, photography, how high

**t_n=art**

**Title:** solves like math theory

weight up on tools, solve your skills
it's skill to need and to assume
she’s over-bored and self-assured
oh no, i work a finite phrase
hello, hello, hello, how high ...
with the lamps out, it’s less difficult
here we are still, interrupt us
i sense greedy and infectious
here we are still, interrupt us
a stochastic, a regression, a prevention, my cognition

**t_n=computational**

**Title:** utilizes like modeling creativity

computation up on rifles, solve your colleagues
it's mathematics to need and to underestimate
she's over-bored and self-assured
oh no, i understand a computational translation
hello, hello, hello, how theoretical ...
with the algorithms out, it's less mathematical
here we are still, enlighten us
i work conceptual and probabilistic
here we are still, enlighten us
a dynamical, a neuroscience, an epidemiology, my optimization

**t_n=art**

**Title:** teaches like art museum

design up on arts, work your artists
it's exhibition to take and to teach
she's over-bored and self-assured
oh no, i think a contemporary literature
hello, hello, hello, how high ...
with the paintings out, it's less known
here we are still, participate us
i want naive and infectious
here we are still, participate us
a conceptual, a curator, an exhibit, my photography

Figure 4: Lyrics for STLT with the History constraint.

as \( w_o \), otherwise, test the following most similar candidate.
For SLTS, this results in the lyrics of figure 5.

**t_n=computational**

**Title:** utilizes like modeling creativity

computation up on rifles, solve your colleagues
it's mathematics to need and to underestimate
she's over-bored and self-assured
oh no, i understand a computational translation
hello, hello, hello, how theoretical ...
with the algorithms out, it's less mathematical
here we are still, enlighten us
i work conceptual and probabilistic
here we are still, enlighten us
a dynamical, a neuroscience, an epidemiology, my optimization

**t_n=art**

**Title:** teaches like art museum

design up on arts, work your artists
it's exhibition to take and to teach
she's over-bored and self-assured
oh no, i think a contemporary literature
hello, hello, hello, how high ...
with the paintings out, it's less known
here we are still, participate us
i want naive and infectious
here we are still, participate us
a conceptual, a curator, an exhibit, my photography

Figure 5: Lyrics for STLT with the PoS constraint.

Although this works for most cases, a limitation arises from the fact that the PoS tagger was trained in sentences of the Wall Street Journal: besides being very different from the style of lyrics, the lines of the latter rarely correspond to complete sentences. Therefore, a minority of grammatical issues is still expected.

**Considering the Metre**

With previous fixes, new lyrics can be produced on the new theme, also meeting grammatical constraints. Yet, several results are hard to sing in the rhythm of the original song melodies. To improve this, in addition to the previous constraints, selected words \( w_n \) have to match the metre of the original words \( w_o \). More precisely, each \( w_n \) must have the same number of syllables and the position of its primary stress must coincide with the target \( w_o \). This information can be acquired from the CMU Pronouncing Dictionary\(^6\).

As the lyrics in figure 6 show, with this constrain, text is easier to sing.

**t_n=computational**

**Title:** solves like math theory

weight up on tools, solve your skills
it's skill to need and to assume
she’s over-bored and self-assured
oh no, i work a finite phrase
hello, hello, hello, how high ...
with the lamps out, it’s less difficult
here we are still, interrupt us
i sense greedy and infectious
here we are still, interrupt us
a stochastic, a regression, a prevention, my cognition

**t_n=art**

**Title:** writes like art culture

weight up on arts, work your works
it’s dance to take and to afford
she’s over-bored and self-assured
oh no, i think a public name
hello, hello, hello, how high ...
with the shows out, it’s less serious
here we are still, introduce us
i want crazy and infectious
here we are still, introduce us
an artistic, a curator, a museum, my exhibit

Figure 6: Lyrics for STLT with the Metre constraints.

**Considering Rhymes**

Beyond metre, a final constraint concerned rhymes. One possibility, perhaps the most natural, would be to guarantee that pairs of words that rhymed in the original lyrics still rhyme in the new. However, in order to better resemble the sound of the original song, also because it was easier, we opted to constrain \( w_n \) such that it rhymes with the target \( w_o \). This was achieved by selecting \( w_n \) only if its termination has the same sound as \( w_o \)’s again according to the CMU Dictionary. For SLTS, this results in the lyrics of figure 7.

The first impression is that, although the new words rhyme, many end up not being changed, because no word with the required termination is found in the most similar 2,500. This has a negative impact on novelty (i.e., many words are the same as in the original lyrics) and relatedness with the new theme \( t_n \) is low. To analyse the impact of testing more words, the maximum number of similar words was set to 100,000, with results in figure 8.

Although more words are indeed replaced, the topic is still too distant from \( t_n \). With 2,500 similar words, few \( w_n \) are replaced, thus not shifting the theme enough to \( t_n \), but with 100,000, many words are replaced by others that are not that clearly related to \( t_n \), and definitely not an analogy of the desired kind. In fact, this does not happen only when the rhymes constraint is added. The analysis of these and other results confirmed that adding constraints has a positive impact on coherence and metre, but the relation of words \( w_n \) to \( t_n \) is also gradually weaker. Observation also suggested that a good equilibrium between coherence and coherence

\(^6\) [http://www.speech.cs.cmu.edu/cgi-bin/cmudict](http://www.speech.cs.cmu.edu/cgi-bin/cmudict)
relatedness is achieved with all constraints but the rhymes, which is further analysed in the following section.

Due to space limitations, we cannot show many resulting lyrics, but resulting titles give a good idea of what happens for different values of $t_n$. See Table 1 for a selection of titles obtained with the Metre constraints.

**Evaluation**

Given the underlying subjectivity, in order to confirm our initial conclusions, we relied on the opinion of humans. For this purpose, we prepared a survey for the assessment of related aspects of the resulting lyrics, namely novelty, grammar, semantics, singability and overall appreciation. The questions of the survey were uploaded to Amazon Mechanical Turk\(^7\) with the following instructions:

- **Summary:** Answer the following questions regarding the proposed new lyrics, considering popular songs and their original lyrics. Thank you very much for your help!

- **Detailed Instructions:** (i) Recall the following song and, if you need, listen to it, e.g., on Youtube (URL); (ii) Read the new proposed lyrics and answer the following questions on different aspects; (iii) The meaning of the slider values is: 1—Strongly disagree, 2—Disagree, 3—Neutral, 4—Agree, 5—Strongly agree; (iv) All answers are mandatory.

Figure 9 is an example of an assignment, which comprised seven questions, aiming to assess selected aspects. Six of those questions were to be answered with a 5-point Likert scale, namely asking: (i) Whether the judge was familiar with the song, which would later enable to ignore answers by unfamiliar users; (ii) How different the new lyrics were to the original (roughly, novelty towards the original song); (iii) How grammatical were the new lyrics (grammaticality); (iv) How semantically coherent the new lyrics were {semantics}; (v) How easy it was to sing the new lyrics with the melody of the original lyrics (singability); (vi) What was the overall appreciation of the new lyrics (overall). The fifth question asked the judge to select the best topic for the song. Given a list of eight words, they had to pick one, or none. This included the six themes used for producing the lyrics in this evaluation – art, computational, eat, elections, sick, sing – plus two additional words – love, war. Our hypothesis is that selecting $t_n$ as the topic is a strong indicator that the new lyrics are indeed about $t_n$.

The aforementioned themes were used for producing lyrics with four different configurations: (i) Keep stopwords + Replacements history (History); (ii) Previous + PoS tagging (PoS); (iii) Previous + Metre (Metre); (iv) Previous + Rhymes (Rhymes). For producing the lyrics with each configuration, the 2,500 most similar words were always tested for replacement. The following original songs were used: (i) Beat it, by Michael Jackson; (ii) Enjoy the Silence, by Depeche Mode; (iii) Heroes, by David Bowie; (iv) Highway to Hell, by AC/DC; (v) Smells Like Teen Spirit, by Nirvana.

Combining the six themes and the five original lyrics, 30 different lyrics were generated for each configuration. As four different configurations were tested, the evaluation set had 120 different lyrics. For each of those, three different judges answered the previous survey, resulting in 360 completed assignments, on which we can rely for comparing the results of each configuration, on each targeted aspect. For each configuration and assessed aspect, figure 2 shows the Mode (Mo) and the Median ($\tilde{x}$) of all aspects, according to the judges. An exception is the topic aspect, which could have multiple answers, but only one was correct. In this case, the table shows the proportion of assignments for which the selected topic was Correct ($= w_n$) or None.

\(^7\)https://www.mturk.com/
Table 1: Titles produced with the Metre constraints.

<table>
<thead>
<tr>
<th>Metre</th>
<th>Original</th>
</tr>
</thead>
<tbody>
<tr>
<td>art</td>
<td>smells like teen spirit</td>
</tr>
<tr>
<td>computational</td>
<td>sculpture to art</td>
</tr>
<tr>
<td>eat</td>
<td>roadway to meal</td>
</tr>
<tr>
<td>elections</td>
<td>roadway to flu</td>
</tr>
<tr>
<td>sick</td>
<td>freeways to song</td>
</tr>
<tr>
<td>sing</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Mode and Median of the rating different aspects in lyrics produced with different configurations.

<table>
<thead>
<tr>
<th>Aspect</th>
<th>History</th>
<th>PoS</th>
<th>Metre</th>
<th>Rhymes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Novelty</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Grammar</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Semantics</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Topic</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Singability</td>
<td>66%</td>
<td>68%</td>
<td>56%</td>
<td>6%</td>
</tr>
<tr>
<td>Overall</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 9: Example assignment of the evaluation survey.

Table 3: Results of the survey.

Table 4: Overall appreciation.

To a large extent, the results of the survey confirm our initial conclusions. Using only the history of replacements leads to higher novelty, because more words are replaced. On the other hand, singability is clearly the lowest, as well as the overall appreciation. This is also the configuration with the least coherent semantics. Though surprising, because words selected this way should be the closest to an actual analogy, this aspect can be indirectly affected by the gram-
matical issues. When adding the PoS constraint, novelty is comparable, and so is grammaticality, which suggests that it is not clear that considering the PoS improves the syntax. But semantic coherence is better, which can be an indirect consequence of better syntax. Singability is also improved, but overall appreciation is the same as in the previous configuration. Both of these lead to the best proportion of assignments with the correct topic, respectively 66% and 68%.

As expected, introducing the metre constraint leads to lower novelty, lower proportion of assignments with the correct topic, though still positive, and improvements in all the other aspects, visible on the higher mode. Where this configuration stands out, is for having the best overall appreciation. Finally, when adding the rhymes constraint, novelty, grammar and the overall appreciation have a slight decrease, whereas semantics and singability improve. The main drawback of adding this constraint is that the proportion of correct topics is very low, even lower than the random chance (11%). Our interpretation is that this is due to the low number of replaced words, which results in lyrics very close to the original, thus easy to sing and with similar semantic cohesion, but in a very different topic than \( t_n \).

On the familiarity with the songs, both mode and median were always 4 or 5. For the History and Rhymes configuration, six judges answered this question with 1 or 2, and nine for the other two configurations. Yet, if we ignore such answers, the only change is that the mode of the grammar aspect for the History configuration drops to 1.

For a broader idea of the results produced by WAM, figure 10 shows three lyrics for which the mean overall appreciation was 4 or higher, all generated with the Metre configuration. Figure 11 shows three lyrics with overall appreciation between 1 and 2, produced with different configurations. Curiously, although two judges rated the third with 1, another rated it with 4. Despite the rating differences, in the third example of each figure, an issue occurs when replacing the original token ‘don’t’. The tokenizer splits it into ‘do-n’t’, but only ‘do’ is replaced, resulting in odd constructions like wants n’t and eats n’t. Lyrics with higher appreciation are slightly further from \( t_n \), but still, to a great extent, semantically coherent. On the other hand, singability of lyrics by the PoS and History is confirmed to be low. For the latter, grammatical issues and strange words (e.g., wo) also contribute to the lower rating.

Conclusion

Interesting results were achieved, which confirmed that we can indeed rely on the analogy operation in word embeddings for automatically shifting the meaning of a poem, as long as some constraints are considered. We illustrated the impact of such constraints in WAM, a system that relies on analogy for transforming lyrics according to a new theme. Moreover, towards better appreciation, human opinions confirmed that replacement words should also have the same PoS and metre as the original. WAM can be seen as a fast way of generating parodies, or even advertising campaigns, based on original songs, poetry, or even other kinds of text as well (e.g., news headlines). To fix still existing issues, results may always be further curated.

Although we are generally happy with the results, in the future, additional experiments can still be performed at different levels. For instance, different priorities can be set for different constraints (e.g., if it is not possible to meet them, drop low-priority constraints); a language model can be used for considering the replacement given the previous or next word(s); or alternative analogy solving methods (e.g., (Levy and Goldberg 2014)) can be tested. To access future changes, we could possibly automate the evaluation of some aspects (e.g., novelty with ROUGE) and follow alternative ways for evaluating others. On the latter, we noticed that, in some cases, the topic was incorrect, but somehow related to \( t_n \) (e.g., sick instead of eat). To minimise this, the topic might become an open answer and we may rely on its similarity with \( t_n \) for assessing its suitability.

We should add that, although we worked with English, a similar approach could be followed for transforming lyrics in other languages, as long as there is a model of word embeddings, a list of stopwords, a PoS tagger, and a method of splitting syllables and identifying the stress of words.

References


Hämäläinen, M., and Alnajjar, K. 2019b. Let’s FACE it. Finnish poetry generation with aesthetics and framing. In
Title: Expect the Ballot
polls like balloting
vote the ballot
go coming in
into my even year
crucial to me
pierce back through me
can’t you represent?
oh my even boy
all i ever promised
all i ever needed
is here in my votes
polls are very unnecessary
they can only do threat

Title: Roadway to Flu
caring healthy, caring sick
rookie patient on an one-way train
telling reason, need me be
treating everything in my gait
don’t care worry, don’t care treat
ain’t reason i would rather do
going down, labour week
my pets are gonna be there too
i’m on the roadway to flu
on the roadway to flu
roadway to flu
i’m on the roadway to flu

Title: Vote it
they said him do n’t you ever want around here
don’t say to change your fear, you even reappear
the conflict’s in their polls and their terms are really stressed
so vote it, now vote it
you even own, you even do what you can
don’t say to change no rule, do n’t be a midterm role
you say to be strict, even do what you can
so vote it, but you say to be due
now vote it, vote it
no part plans to be elected
showin’ how quirky and weak is your war
it wants n’t issue who’s clear or left
now vote it, vote it

Figure 10: Lyrics with high overall appreciation, all with Metre configuration: Enjoy the Silence with \( t_n = \text{elections} \); Highway to Hell with \( t_n = \text{sick} \); Beat it with \( t_n = \text{elections} \).

Figure 11: Lyrics with low overall appreciation: Enjoy the Silence with \( t_n = \text{eat} \) and History configuration; Highway to Hell with \( t_n = \text{election} \) and PoS configuration; Beat it with \( t_n = \text{eat} \) and Metre configuration.
Towards balanced tunes: A review of symbolic music representations and their hierarchical modeling

Nádia Carvalho
University of Porto, Faculty of Engineering
Porto, Portugal
ei12047@fe.up.pt

Gilberto Bernardes
INESC TEC
University of Porto, Faculty of Engineering
Porto, Portugal
gba@fe.up.pt

Abstract
Since the heydays of music informatics, around the 1950s, the modeling and prediction of musical structures manifested as symbolic representations have been continuously pursued. The operational property of such methods is to provide the conditional distribution over an alphabet – i.e., the entire collection of unique musical events in a composition or corpus – given a context – i.e., a preceding sequence. This distribution unpacks temporal morphologies that support multiple applications for predictive and assisted creative tasks, such as the generation of new musical sequences that retain a structural resemblance to a modeled source. Despite their longstanding tradition, state-of-the-art methodologies for symbolic music modeling are yet to reach the music community. Naive models such as Markov chains, which are known to neglect the fundamental hierarchical nature of musical structure, remain common practice. In this paper, we extensively review existing methodologies for symbolic music representation and modeling, as the first steps towards a study on the resulting balance across familiarity and novelty in generative music applications.

Introduction
Historically, music informatics has been exploring the algorithmic modeling and prediction of musical structure. Existing applications stem from information theory principles and the postulate of music as a low entropy phenomenon (Conklin and Witten, 1995). Given the temporal and hierarchical nature of the musical structure, algorithmic methods are typically informed by a sequence of past events, i.e., a context, to both model existing structures and predict or generate new structures (Conklin and Anagnostopoulou, 2001). These models aim to capture different degrees of inter-dependency across the component elements of the musical structure. Prior to the modeling of musical structure, a discrete and finite alphabet including all unique symbolic representations for a given structure has to be created. Depending on the adopted intra- and inter-opus musical material, algorithmic models capture different musical traits ranging from recurrent patterns in a composition to stylistic idiosyncrasies of a composer or even tonal music principles.

The balance between familiarity to known compositional traits, captured by these algorithmic methods and novelty introduced by unfamiliar and unpredictable structures is of utter importance in the design of generative systems (Bevington and Knox, 2014). The Wundt curve, a hedonic function that relates the levels of novelty and expectation to the ‘pleasantness’ of creative works (Berlyne, 1970), captures the notion of balance as mentioned above.

In this paper, we argue that the interaction between discrete and finite alphabets of music and their temporal modeling is instrumental in controlling the resulting balanced of generative music models across the novelty-familiarity range. To this end, we extensive review musical representations and modeling methods adopted in the context of generative music, as the first steps towards a larger study on their (balanced) interaction thereof.

The remainder of this paper is structured as follows. Section “Symbolic Representation of Musical Structures” provides a literature review on the topic. Section “Modeling Temporal Musical Structures” presents modeling methods that capture the morphology of musical structures. Section “Applications” reviews representative generative music applications, which combine the two above components and have a broader adoption by the music community. A twofold categorization of computer-aided algorithmic composition and machine improvisation applications is adopted. Finally, Section “Summary and Future Challenges” presents the conclusions and discusses future challenges.

Symbolic Representation of Musical Structures
In this section, we review the following four symbolic music representations adopted in the computational modeling of musical structure: formal strings, graphs, formal grammars, and geometrical representations. These representations were selected based on their focus, relevancy and impact in improving the models for musical structure modeling and prediction across related literature.

Formal Strings
Formal strings are one of the earliest and most frequently adopted computational representations of symbolic music manifestations. It encodes musical structure as sequences of symbols driven from a finite and discrete alphabet, \( \Sigma \). To encode duple pitch-duration information – two primary elements in Western music (Wishart and Emmerson, 1996)
Table 1: Multiple encoding of pitch and duration using formal string representations for the musical excerpt shown in Figure 1.

<table>
<thead>
<tr>
<th>Pitch/Duration Encoding</th>
<th>Encoded Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common music notation ( p_{cmn} )</td>
<td>( (G,^1,3) ) ( (G,^1,3) ) ( (D,^1,3) ) ( (G,^1,3) ) ( (B,^1,3) ) ( (C,^1,4) ) ( (D,^1,4) ) ( (B,^1,3) ) ( (A,^1,3) )</td>
</tr>
<tr>
<td>Absolute Pitch (MIDI values)</td>
<td>55 55 50 43 59 60 62 60 59 57</td>
</tr>
<tr>
<td>Base-12 ( p_{12} )</td>
<td>8 8 3 8 12 13 1 12 10</td>
</tr>
<tr>
<td>Base-21 ( p_{21} )</td>
<td>13 13 4 13 19 14 1 19 16</td>
</tr>
<tr>
<td>Base-40 ( p_{40} )</td>
<td>26 26 9 26 38 39 3 38 32</td>
</tr>
<tr>
<td>Interval ( \tau_{dc} )</td>
<td>0 0 -5 -7 16 1 2 -2 -1 -2</td>
</tr>
<tr>
<td>Interval from tonic ( \tau_p )</td>
<td>0 0 7 0 4 5 7 5 4 2</td>
</tr>
<tr>
<td>Contour ( \tau_c )</td>
<td>0 0 -1 -1 1 1 1 -1 -1 -1</td>
</tr>
<tr>
<td>HD-Contour ( \tau_{hd} )</td>
<td>0 0 -3 -3 4 1 1 -1 -1 -1</td>
</tr>
<tr>
<td>Absolute time ( \tau_{taba} )</td>
<td>0 1/2 1 3/2 2 9/4 5/2 11/4 3 13/4</td>
</tr>
<tr>
<td>Absolute duration ( \tau_{daba} )</td>
<td>1/2 1/2 1/4 1/4 1/4 1/4 1/4 1/4 1/2</td>
</tr>
<tr>
<td>Contour ( \tau_c )</td>
<td>0 0 0 0 -1 0 0 0 0 0</td>
</tr>
<tr>
<td>HD-Contour ( \tau_{hd} )</td>
<td>In this case, the resulting string is the same as ( \tau_c ) because there is only changes between close rhythm durations.</td>
</tr>
</tbody>
</table>

Table 2: Some basic and derived viewpoints for the events of Figure 1.

<table>
<thead>
<tr>
<th>Type</th>
<th>( e_1 )</th>
<th>( e_2 )</th>
<th>( e_3 )</th>
<th>( e_4 )</th>
<th>( e_5 )</th>
<th>( e_6 )</th>
<th>( e_7 )</th>
<th>( e_8 )</th>
<th>( e_9 )</th>
<th>( e_{10} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>start offset</td>
<td>10</td>
<td>12</td>
<td>14</td>
<td>16</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>25</td>
</tr>
<tr>
<td>pitch (absolute)</td>
<td>55</td>
<td>55</td>
<td>50</td>
<td>43</td>
<td>59</td>
<td>60</td>
<td>62</td>
<td>60</td>
<td>59</td>
<td>57</td>
</tr>
<tr>
<td>duration</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>key signature</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>time signature</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>delast (is rest?)</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>posbar (position in bar)</td>
<td>10</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>fb (is first in bar?)</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>sequit (sequential interval from last note)</td>
<td>( \bot ) 0</td>
<td>-5</td>
<td>-7</td>
<td>16</td>
<td>1</td>
<td>2</td>
<td>-2</td>
<td>-1</td>
<td>-2</td>
<td></td>
</tr>
<tr>
<td>contour</td>
<td>( \bot )</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>hdist</td>
<td>( \bot )</td>
<td>0</td>
<td>-3</td>
<td>-3</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>referent</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>thrb (semit in bars)</td>
<td>( \bot )</td>
<td>0</td>
<td>( \bot )</td>
<td>( \bot )</td>
<td>( \bot )</td>
<td>( \bot )</td>
<td>( \bot )</td>
<td>( \bot )</td>
<td>( \bot )</td>
<td>( \bot )</td>
</tr>
<tr>
<td>thrq (semit at quarters)</td>
<td>( \bot )</td>
<td>0</td>
<td>( \bot )</td>
<td>-12</td>
<td>( \bot )</td>
<td>-19</td>
<td>( \bot )</td>
<td>( \bot )</td>
<td>( \bot )</td>
<td>( \bot )</td>
</tr>
</tbody>
</table>

Figure 1: The first bar of J. S. Bach’s Courante of Suite No. 1 in G major, BWV 1007.

The multiple viewpoint systems (Conklin and Witten, 1995) emerged as an extension of the dupl pitch-duration formal string representations. It expands the former formal string representations by including secondary structural information, such as metrical position and interval. These systems use domain knowledge to derive new representations for encoding temporal events from the musical structure by abstracting properties types, \( \tau \), as summarized in Table 1. To compute each type a function \( \Psi \) is adopted. A viewpoint comprises one such function and the set of strings that can be computed. A multiple viewpoint system comprises a collection of different viewpoints, some of which can be derived from basic viewpoints. Furthermore, as the viewpoints can have correlations, a new type was introduced: the product type \( \tau_x \otimes \tau_p \), whose elements are the cross product of their constituents. Table 2 shows a set of multiple viewpoints of the musical excerpt in Figure 1.

Polyphonic (i.e., multi-layer) formal string representations can be split into the following three categories: non-interleaved (Lemström and Tarhio, 2003), interleaved (Pienimäki, 2002) and onset-based (Lemström and Tarhio, 2003). The first category encodes polyphonic music textures as independent monophonic layers sequentially. The second category encodes all layers linearly, ordering pitch values sequentially by their onset times. It provides greater flexibility for handling the complex multidimensional nature of polyphonic music; however, it does not highlight the vertical, namely homophonic, nature of the textures. The third category underlines homophonic textures by discriminating all overlapping notes as vertical aggregates. However, the duration information is lost and it can lead to a combinatorial explosion. Hanna et al. (2008) minimize the lack of duration information by fragmenting long notes into notes of fixed duration connected by ties.

The task of capturing all co-dependencies across vertical and horizontal textures is still challenging and has not yet been fully achieved.

**Graphs**

**E-Graph** is a representation of monophonic sequences proposed by Marsden (2001). It typically adopts a minimum of two places (i.e., nodes) for each symbol, namely time and pitch.

Places can be connected by elaborations (i.e., edges), which typically include metrical and pitch information. The latter generates new intermediate places without crossing links, making the representation interpretable as an acyclic graph, hence easily represented as a tree. Elaborations can either be simple or accented. Simple elaborations refer to insertions between two-note events, such as rests, repetitions, anticipations, passing notes, and octave jumps. Accented elaborations refer to delays, suspensions, and accented passing notes.

E-Graphs have shown great potential in capturing musical patterns in multiple stylistic contexts. However, it was gradually abandoned due to its: i) excessive complexity, ii) am-
Tagliolato (2008) presented a time-independent graph-based signature of melodic layers as an alternative to E-Graphs. It adopts a reduced 12-tone pitch alphabet, which captures invariant-pitch structures under inversion and retrogradation. Basic rhythmic features and events’ order can be encoded in the graph’s edges.

**Formal Grammars**

Formal music grammars, or simply grammars, represent the “intuitions of a listener who is experienced in a musical idiom” (Lerdahl and Jackendoff, 1983, p. 3) using formalized methods (i.e., higher-level abstractions). It can be split into four hierarchies: i) grouping into motives, phrases, periods or sections, ii) metric alternation of strong and weak beats, iii) time-span reduction from metrical and grouping structures to higher-level hierarchies, and iv) prolongational reductions describing tension and relaxation phenomena across time (Lerdahl and Jackendoff, 1983). Despite being well-suited for representing structural dependencies in musical structure, formal context-free 1 music grammars are challenging to compute. Their strict hierarchy is difficult to reconcile with the inherent ambiguity of musical structure (Rohrmeier and Pearce, 2018).

Rizo (2010) proposed a formal grammar representation for computing the similarity of monophonic and polyphonic music sequences as tree structures. The tree leaves represent note events and pitch class sets from monophonic and polyphonic layers, respectively. Employing multi-sets with duration information and number of notes overlapping solves the problem of encoding polyphonic information. Rizo showed that tree structures are versatile in encoding information other than pitch or rhythm, such as harmonic structure and form. However, its dependency on a priori knowledge of the metric structure and its high complexity in representing ties, dots, and syncopations are prominent drawbacks.

**Geometric Representations**

Ma`din (1998) proposes the use of geometric representations to encode music as 2-dimensional pitch-duration contours across time. From the resulting geometrical representation, various metrics were proposed, namely for similarity computation across musical sequences.

A particular case of these geometric representations is the multidimensional point sets, proposed by Meredith (2006), that adopt the Euclidean space to represent musical events as a tuple of 5 elements: onset time of the note; chromatic (absolute) pitch; diatonic pitch, defined by an integer that indicates the position of the head of the note on the staff; duration; and the voice (i.e., layer) number within the polyphonic texture. The 2- or 3-dimensional projection of the

\[ ... \]

1A context-free formal grammar is a set of production rules that describe all possible strings in a given formal language by allowing the application of those rules regardless of the context of the nonterminal symbols.

point sets allows the efficient search for similar patterns, including small variations, using their spatial configuration. Figure 2 shows a projection of onset time and chromatic pitch for the musical excerpt in Figure 1.

![Figure 2: A Multidimensional Point Sets projection of onset time and chromatic pitch for the example in Figure 1.](image)

**Modeling Temporal Musical Structures**

In this section, we review the following two musical structure modeling techniques, which were selected based on the visibility and attention they attract from the music community: statistical sequence modeling and compression algorithms. Despite the possibility to manually draw these models from scratch, they are typically driven from existing musical structures (e.g., individual pieces or a corpus). Despite falling into the general category of musical structure modeling, we do not include evolutionary computation algorithms in this review, as they do not address domain-specific knowledge (Nierhaus, 2009).

**Statistical Sequence Modeling**

*N*-grams are specific types of Markov models, which capture dependencies across discrete and finite symbols from an alphabet, given a context (Downie, 1999). *N* corresponds to the total number of contiguous symbols under consideration in the model, i.e., the context. Despite its popularity, *N*-grams have been criticized due to their limitation in capturing long-term structural dependencies (Sears et al., 2017).

When encoding musical structures in multiple directions or hierarchies, the number of associations between events can explode in combinatorial as *N* and the length of the original sequence increase (Sears et al., 2017).

In light of this limitation, *skip*-grams have been proposed to parse non-contiguous elements from the musical structure. The maximum length of these skips can be defined by a threshold in the fixed-skip model. Symbols are only considered if within a fixed range of skips from the event processed. Alternatively, it can follow a variable-skip approach that parses all events satisfying a particular condition (Herrmanns and Chuan, 2017). The latter approach is typically adopted for modeling temporal-dependent sequences.

Sears et al. (2017) has shown that *skip*-grams significantly outperform contiguous *n*-grams in discovering cadences. Markov chains embed *n*-gram and *skip*-grams models to generate musical sequences that are statistically similar to modeled sources.
Factor Oracle (FO) was introduced by Allauzen, Crochemore, and Raffinot (1999) as acyclic automata that recognizes at least the factors of a word. FO is a time- and memory-efficient string-matching algorithm and has been recently proposed for modeling musical structures.

FO is learned online in an incremental fashion. Repeated patterns in FO are denoted by two types of links between states: factor links and suffix links. Factor links indicate paths across states that produce similar patterns by continuing forward. Suffix links indicate paths across states that share the largest similar subsequence from the input sequence. FO is particularly useful in satisfying the incremental and fast online learning, time-bounded generation of musical sequences, and implementation of multi-attribute models to deal with the multi-dimensionality of music (Tatar and Pasquier, 2019).

Toro (2016) and Dégueurnel, Vincent, and Assayag (2018) extend FOs towards the introduction of link probabilities to maximize novelty when adopting the model for generative purposes. Furthermore, they promote the application of FO to multidimensional domains such as polyphonic music or improvisations with multiple musicians.

The Variable Markov Oracle (VMO) was proposed by Wang and Dubnov (2014a) for clustering multivariate time series without a priori assumptions on the number of clusters. VMO algorithm is based on FO (Allauzen, Crochemore, and Raffinot, 1999) and Audio Oracle (Dubnov, Assayag, and Cont, 2007). It allows the construction of the oracle without an initial alphabet. To this end, it introduces a threshold variable for computing the degree of similarity across states. The threshold value in VMO typically adopts an entropy metric to capture the information rate across events (Wang and Dubnov, 2014b).

Wang, Hsu, and Dubnov (2016) made a first attempt at establishing a statistical model for VMO by making an analogy to the HMMs based on the inference of emission probabilities, without introducing probabilities to the transitions themselves. Transition probabilities in the VMO were later proposed by Wang and Dubnov (2017), using the lengths of longest repeated suffixes, which provide variable-length Markov transition information. This new model has shown to provide a more compact and abstract representation of the oracle structure while keeping its variable-length Markov properties. Furthermore, it allows the processing of multiple works (i.e., a corpus) in a single VMO.

Hidden Markov Models (HMMs) capture relations between states that are partially hidden, i.e., unknown. Probability distributions per state define a particular alphabet symbol emission. These models are defined by a tuple of five elements that correspond to: i) the finite alphabet of visible symbols, ii) the finite set of states, iii) the mappings defining the probability of transitions between hidden states, iv) the emission probability of each visible symbol at a given hidden state, and v) the initial probabilities of the hidden states. HMMs can process complex structures of sequential data but require a considerable understanding of the problem domain and a large number of training examples (Schulze and van der Merwe, 2011).

Frankel-Goldwater (2007) implements HMM using the forward-backward, Viterbi, and Baum-Welch algorithms for modeling pitch, duration and dynamics from musical structures. Schulze and van der Merwe (2011) computes the parameters for HMMs of variable order by means of empirical counts to capture monophonic arc structures and accompanying chord progressions.

Compression Algorithms

General-purpose lossless compression algorithms use the redundancy of input sequences to decrease storage memory size while maintaining the information in full. When applied to musical structures, these algorithms find relevant patterns and efficiently model musical structures as the “shortest descriptions of any musical object, [...] that describe the best possible explanations for the structure of that object” (Louboutin and Meredith, 2016, p. 2). In this context, we will review the following compression algorithms adopted in music structure modeling: LZ77 and LZ78, the Burrows-Wheeler Transform and the variants in the Structure Induction Algorithm (SIA) family.

Ziv and Lempel’s LZ77 (1977) and LZ78 (1978) are two of the most popular lossless data-compression algorithms. Both algorithms adopt a dictionary-form of the alphabet from the original musical structure to be compressed. LZ77 replaces portions of the input data with symbols representing the longest found match, in run-length encoding format, using a sliding window. The larger the window, the highest amount of recent data is acquired. The encoder can too search farthest back for creating references. LZ78 improves the performance of LZ77 by using an ordered dictionary of reusable data and its indexes, instead of the actual stream data.

The Transform of Burrows and Wheeler (1994) uses a suffix array to permute the input data structure so that identical elements are brought closer together. It increases the probability of finding an event from an alphabet if there are near occurrences of the same event. Along with move-to-front coding, it builds the alphabet from the events in the structure using left to right parsing and constructs a vector of the alphabet indexes, which promotes enhanced compression factors.

The family of Structure Induction Algorithms (SIA) aims at discovering maximal repeated patterns from n-dimensional sets of points in Cartesian spaces, namely those representing musical structures (Meredith, Wiggins, and Lemström, 2002). SIA discovers all maximal subsets from a n-dimensional point set with an ordering metric and removes repetition under symmetry. SIATEC (Meredith, Wiggins, and Lemström, 2002) extends SIA by finding all occurrences of maximal repeated patterns, including those related by translational equivalence as a Translational Equivalent Class (TEC). COSIATEC (“Compression with SIATEC”) extracts the TECs resulting from SIATEC and selects those that provide “best” compression factor without overlap. RECURSIA-RRT stands for recursive translatable point-set pattern discovery with the removal of redundant
translators. It optimizes the previous algorithms by increasing the compression factor.

Louboutin and Meredith (2016) compares all these algorithms on classifying folk song melodies using a multiple viewpoints system representation. LZ77 and COSIATEC were shown to achieve the best results.

Applications
In this section, we review applications that make use of the representations and modeling techniques detailed in Sections “Symbolic Representation of Musical Structures” and “Modeling Temporal Musical Structures”. The non-comprehensive, yet representative, selection was based on the visibility and attention the applications attract from the music community. We adopt the following twofold categorization of the applications: computer-aided algorithmic composition (CAAC) and machine improvisation.

Computer-aided algorithmic composition
CAAC systems refer to computer applications that promote the generation of musical structures by means other than the direct manipulation the musical surface elements (Ariza, 2005a). These computational systems expand compositional design strategies towards the adoption of (semi-)automatic algorithmic techniques at different levels of the composition. A process referred to as meta-composition (Ariza, 2005b).

One of the earliest algorithmic composition examples is the Illiac Suite by Hiller and Isaacson (1957). It adopts rule-based systems and Markov chains to compose formal music structures. Inspired by Hiller and Isaacson’s work, Baker proposed in 1963 the library MUSICOMP, which implements its various algorithmic composition methods (Ames, 1987).

In the early-1960s, Xenakis, renowned for his stochastic processes, uses computers to automate his composition methods (Ames, 1987). Koenig, another pioneer, implements some techniques, such as Markov chains, to automate the generation of music structures (Ames, 1987). Berg (1995) lately compiled these techniques in a collection of tools, the AC Toolbox, to promote various methods for algorithmic composition.

In 1981, Cope presented EMI (Experiments in Musical Intelligence). The system learns stylistic traits from a music corpus, manifested in the MIDI standard, to imprint them into generated musical structures (Cope, 1989). The musical information at the note level is encoded in a multidimensional point set, although the term had not yet been coined. The relation of these musical units is encoded in a formal grammar.

CACIE (Computer Aided Composition using Interactive Evolution) is an application by Daichi Ando and Hitoshi Iba (2007) that aims to assist composers in creating atonal music. It uses formal grammars to represent musical phrases from music manifested in the MIDI standard and an evolutionary (genetic) system to generate new musical structures.

FlowComposer is an interactive music composition environment developed by Papadopoulos, Roy, and Pachet (2016). It uses Markov chains to automatically compose lead sheets, which are further harmonized by style-specific traits encoded in formal string representations driven from MIDI musical corpora.

Morpheus uses the COSIATEC pattern recognition technique to find repeated sequences in a musical piece, in MIDI format, combined with a three-dimensional geometric model (the Spiral Array) with tonal tension information from MusicXML music. Found sequences are then used to constraint generative polyphonic music processes based on evolutionary computation (Herremans and Chew, 2016).

Machine improvisation
Machine improvisation refers to musical collaborations between humans and machines in an improvisation setting. Lewis’ Voyager (1988) is a pioneer work that expands upon the concept of “virtual improvising orchestra.” It produces variations from live performers’ MIDI input data and generates responses accordingly, using a rule-based, formal-grammar approach (Lewis, 2000).

The Continuator was proposed by Pachet (2003) as a system that “bridges the gap between interactive musical systems, limited in their ability to generate stylistically consistent material, and music imitation systems, which are fundamentally not interactive.” It adopts variable-length Markov chains to model MIDI input data from a live musician encoded as a tree structure. A weighted fitness function controls the level of ‘sensitivity’ to the musical context.

OMax is a real-time system presented in 1998 and under active development. The system learns the style of live musicians and actively participates in an ongoing performance as a co-improviser machine. OMax uses the FO, and lately the VMO, to learn stylistic traits from a performer’s MIDI stream in the form of formal string encodings.

FILTER (Freely Improvising, Learning and Transforming Evolutionary Recombination) by Nort, Oliveros, and Braasch (2013), combines FO and HMM in a context of free improvisation to learn temporal structures from the input. Moreover, it adopts a fitness function for controlling the level of imitation vs. novelty of the responses.

Summary and Future Challenges
This paper reviewed symbolic music representations using finite and discrete alphabets and temporal modeling techniques that capture musical structure hierarchies.

In Section “Symbolic Representation of Musical Structures”, we detailed multiple strategies to represent musical structures, namely formal strings, formal grammars, graphs, and geometrical representations. These representations can encode complex and hierarchical music structures, yet, only a few can adequately parse inter-part (polyphonic) dependencies, such as multiple viewpoint systems. Encoding linear (i.e., part) and vertical (i.e., inter-part) dependencies should be further explored. Moreover, as noted in Section “Applications”, despite the considerable number of existing representations for musical structure, their adoption by the music community focus almost exclusively on formal string representations. We believe that domain-specific representations, such as the multiple
viewpoint systems, would allow for enhanced control over which structural elements are further modeled.

Methodologies for modeling the temporal structure of music at multiple hierarchies typically draw on pattern recognition methods, such as the LZ77 and LZ78 compression algorithms, and statistical modeling that capture repeating sequences in the longer-term musical structure. The compression algorithms explore exact matches, while statistical modeling, namely the Variable Order Markov Oracles, also contemplate variations (e.g., note insertions, passing notes). Despite the advances on the state-of-the-art, existing modeling techniques inadequately account for implicit, yet important, elements of musical structure, such as phrase boundaries.

In Section “Applications”, we explore the combination of representations and temporal modeling in the scope of CAAC and machine improvisation. These applications are ideal test-beds for exploring the balance across the novelty-familiarity range in generated musical structures. CAAC systems typically require a wider exploration of this range, while machine improvisation tend to rather focus on balanced outputs with greater tendency towards familiarity. Yet, both use predominantly Markov chains, which neither optimize this balance nor provide fine degrees of control across the familiarity-novelty range. VMO shows finer and more flexible degrees of control over the representation and pattern recognition. However, its full capacity within generative music contexts is yet to be explored, namely the non-linear relations between the representation and modeling.

Currently, the state-of-the-art in temporal music modeling is at a crossroads. The rise of deep learning techniques – which can be explained by the increasing amount of available data, and efficient and affordable computing power – can render traditional modeling and prediction obsolete. Representative models are BachBot (Liang et al., 2017) and Music Transformer (Huang et al., 2018). However, a relevant problem of these models is that they often fail to capture the intrinsic non-linear relationships of creative tasks (Briot, Hadjeres, and Pachet, 2019). Deep learning architectures rely on multiple layers to directly extract relevant features from the sources before modeling. Their hyper-specialization towards a particular objective or a specific training corpus and their lack of explainability pose a critical problem to the creative balance. A greater understanding of latent spaces is instrumental in promoting balanced outputs that match user preferences across novelty and familiarity.

References


3Latent spaces comprises the representation of compressed data in deep learning techniques.


EMILY: An Emily Dickinson Machine

Juliana Shihadeh and Margareta Ackerman
Department of Computer Science and Engineering
Santa Clara University, Santa Clara, California

Abstract
This paper introduces EMILY, a machine that aims to create original poems in the style of renowned poet Emily Dickinson. Dickinson’s succinct and syntactically distinct style with unconventional punctuation makes for an interesting challenge for automated poetry creation. A user study compares EMILY’s poems to Emily Dickinson originals, demonstrating the machine’s ability to evoke mental images and highlighting challenges for future work.

Introduction
Poetry writing is an artform dating back to prehistoric times (Finnegan 2012). A successful poem elicits imagery and evokes emotion through an interlock of relationships between semantics, syntax, grammar, punctuation, rhythm and rhyme. Machine generated poetry is itself an artform distinct from human made poetry, with computer generated poems created across human languages through a variety of computing techniques (see, for example, (Lau et al. 2018), (Zhang and Lapata 2014) and (Hämäläinen and Alnajjar 2019)).

While poetry machines often create original works without focus on any particular poet, there are exceptions. Style imitation has, for example, been applied to the works of Italian poet Dante Alighieri (Zugarini, Melacci, and Maggini 2019), Bob Dylan lyrics (Barbieri et al. 2012), and the works of William Shakespeare and Oscar Wilde, amongst several others (Tikhonov and Yamshchikov 2018).

Poetic style imitation offers the opportunity to immortalize a poet by keeping their voice alive through novel works. From an evaluation standpoint, the generated works can be compared with those of the original creator, enabling a variation of the CC Turing Test by checking whether unbiased observers are able to discern generated artifacts from original ones. Other variations involve comparing the original and generated works on important criteria (ex. stylistic elements of poetry) to help identify where improvement is needed.

In this paper, we present EMILY, a poetry machine that aims to replicate the style of Emily Dickinson’s poems. We present the methodology behind EMILY, along with a user study that compares machine-created poems with Emily Dickinson originals on several poetic criteria.

Method
The making of EMILY consists of data preprocessing, the creation of custom Markov Chains, and postprocessing. These steps are detailed below.

Data Preprocessing
EMILY was trained on publicly available Emily Dickinson poetry from the Gutenberg project: “Poems by Emily Dickinson, Three Series, Complete by Emily Dickinson” (Dickinson 2004; Project Gutenberg). The data was made of 444 poems, consisting of 10178 lines.

Punctuation meaningfully contributes to Dickinson’s unique style and as such deserves careful treatment. We saved commas, periods, question marks, and semi-colons. Dickinson is well known for her uses of dashes (Emily Dickinson Museum 2020), which were also preserved. Some
punctuation, particularly all brackets, were omitted, as they introduced noise without helping to capture Dickinson’s style.

Dickinson used to number instead of title most of her poems. We discarded all roman numerals in our preprocessing since our focus is on generating the poems’ bodies.

The final preprocessing step was to convert any fully-capitalized words found in the poem titles into lower case. This helped to enrich the data set of Dickinson’s words. Words that start with capital letters were left unchanged because Dickinson used capitalized words in the middle of sentences (Emily Dickinson Museum 2020).

Custom Markov Chains
To endow EMILY with Dickinson’s style, we chose to build our own custom Markov Chains. This gave us greater control over the creative process, particularly as it pertains to punctuation, which is a central element of Dickinson’s poetry. (Barbieri et al. 2012) also observed that unmodified Markov Chains were insufficient for capturing style, in their case as it pertains to Bob Dylan’s use of rhyming.

The Markov Chains implementation relies on a dictionary. We create the Markov Chains by iterating through all the words and reading them in reverse. Starting with the first word, we iterate for each word at index i checking if the prior word appears in the dictionary. If so, we add the word to its list of values. If the word before it is not in the dictionary, we add it to the dictionary and start its list of values with the current word as the first word. As a result, we map each word to all the words that proceed it in Dickinson’s writing. Doing so lets us capture the relationship of what words show up after each specific word along with their frequency. Words with higher frequency have a higher probability of being generated. Our final dictionary had a total of 8610 keys.

Markov Chains are used to generate the sequence of words for the poems. We format the generated words in the postprocessing phase.

Starting Word For single stanza poems, we randomly select the initial word from all words used in Dickinson’s writing. If the poem has more than one body, we rely on the final word in the previous body in order to generate the first word in the sequence body using the Markov process.

Body Each stanza in a poem is 20 words long. This keeps the poems at approximately the length of Dickinson’s poems, which consist of short stanzas of 4-5 lines each with 5-6 words per line. The number of stanzas generated for each poem is determined by a variable \( n \) passed to EMILY.

Closing Word To help bring out Dickinson’s style, concluding words were chosen from amongst those that had punctuation.

Postprocessing: Formatting the Poems
Not only is the choice of words in the poem important to capturing Emily Dickinson’s style, but the format of the poem brings in important stylistic elements. We format the poems based on an analysis of Dickinson’s poetry.

Dickinson starts poems with capitalized words, and also follows periods, exclamation marks, or question marks with capitalized word. Words that follow a comma or semi-colon are generally lower case. More importantly, Dickinson is known for capitalizing words in the middle of sentences, not only words that begin a new line (Emily Dickinson Museum 2020).

We traverse through the final list of words and set a flag based on the type of punctuation to determine if the following word should start with a capital or lower case letter. Following Dickinson’s style (Emily Dickinson Museum 2020), any capitalized words not preceded by a comma or semi-colon are left unchanged. The generated list of words is then divided into 5 word sentences, and the first letter of each sentence is capitalized.

User Study
We evaluate EMILY by comparing its machine-created poems to Emily Dickinson originals on several criteria. This study seeks to gain an initial understanding on the quality of EMILY’s poems. Larger and more in depth studies are left to future work.

We surveyed 17 participants, 9 female and 8 male. On a scale of 0-5, 0 being “Not at all Familiar” with Emily Dickinson’s poetry and 5 being “Extremely Familiar”, 3 participants responded with a 4, 5 responded with a 3, 4 with a 2, 1 with a 1 and 4 with a 0.

Participants were presented with a total of 12 poems, consisting of 10 of EMILY’s poems and 2 poems by Emily Dickinson. The original poems are Poem 6, “Faith” is a fine invention, and Poem 12, Come Slowly—Eden, which capture many of her stylistic elements.

The choice of questions was influenced by previous work evaluating machine-made poetry (Zugarini, Melacci, and Maggini 2019; Hämäläinen and Alnajjar 2019; Lamb, Brown, and Clarke 2015). For each of the 12 poems, participants were asked the following:

1. Is this a typical poem?
2. Is this poem understandable?
3. How much do you like the word choice in the poem?
4. Does the text evoke mental images?
5. Does the text evoke emotion?
6. Do you like this poem?

Each question was answered by selecting from a Likert scale: Strongly disagree (0), disagree (1), neutral (2), agree (3), strongly agree (4). The scores of each question were averaged across all respondents for each poem, as shown in Figure 2. The scores of each question were also averaged across all generated poems versus the original Emily Dickinson poems, shown in Figure 3.

Results
Our survey shows that question 4, “Does the text evoke mental images?”, had the highest average score of 2.17 of all questions for generated poems. Furthermore, the average score of question 4 outranked the average score for Emily
Dickinson’s poems in 2 of the generated poems. Poem 1, 7, and 10 had the highest score for question 4 as seen in Figure 2. Poems 1, 7 and 10 appear at the end of this section.

Three of our generated poems resulted in at least 3 out of the 5 questions averaging to a score higher than 2, in a range similar to Emily Dickinson’s poems’ average scoring of 2-3 (Poem 1, 3, and 11). Each of these poems performed well on a different set of questions.

The question “Is this poem understandable?” resulted in the lowest average score across all our generated poems as seen in Figure 3, with a score of 1.43 across all generated poems. Dickinson’s poems averaged to a score of 3 on this question. The questions “Is this a typical poem?” and “Do you like this poem?” averaged to 1.72 and 1.76, respectively, identifying areas for improvement.

Most of EMILY’s poems resulted in average scores of around 2. Dickinson’s poems resulted in average scores closer to 3 with question 4 “Does the text evoke mental images?” and question 5 “Does the text evoke emotion?” averaging out to the mid-2s at about 2.71 and 2.6, respectively.

The overall average scores of Emily Dickinson’s poems were higher across all questions compared to our generated poems as seen in Figure 2, which offers an interesting challenge in future work. EMILY’s poems faired well compared to Emily Dickinson’s poems, averaging to a score of about 2 while Dickinson’s averaged to scores closer to 3 with only two in the mid-2s range.

To give the reader a better sense for EMILY’s poetry, we conclude this section with the 3 top performing generated poems in the study.

Poem 1

Some shook their yellow gown
And certainly her eye, they
Leap upon the rose smiling
To die. The orchards Eternity!

Poem 7

The wondrous dear, –An
Enemy is the gate the
Children caper when liked, –
Might but a year, hunted,

Tis all can put out
A little plan to his
Eternal chair, his notice to
Pass odors so dense notoriety.

Poem 10

Surrendering the ‘house at Lexington,
And then of snow; the
Orchard sparkled like perfidy. A
Year, nor heedless were small,

For ’t was to a
Watch, some sweet birds jocoser
Sung; the reason that could
Not put it until mystery!

Comparison to another technique

We compare the results of our custom markov chains model to using a built-in Python markov package, Pypi Markovchain 0.2.5, relying on the same Emily Dickinson poems as the data. Preliminary analysis suggests that our custom method is able to produce poems that capture Emily Dickinson’s

1https://pypi.org/project/markovchain/
Figure 3: Average scores of questions across all EMILY poems versus average scores of questions across original Dickinson poems.

style more closely, with respect to punctuation, formatting, and overall stylistic similarly. Two examples of poems created with the prebuilt Markov Chains are shown below.

Example 1
In the pumpkins in dungeons are known her final inch, chamber and firmaments row of the last included both, danced to see by side, i failed to me.

Example 2
But murmuring of the bewildering thread’s curtain fell, your way soft descent among the sky!

Conclusions

This paper presents EMILY, a machine that aims to create poems in the style of renowned poet Emily Dickinson. Dickinson’s efficient and effective use of words to evoke emotion and imagery, along with distinct syntactic choices, make this an ambitious task, and this paper makes initial steps in this direction. Future work can make further progress guided by the findings of our user study, which highlights areas for improvement.

The initial user study performed here compares EMILY’s poems with original poetry by Emily Dickinson on several dimensions, such as typicality, understandability, and ability to evoke emotion and imagery. The analysis shows that the generated poetry evokes mental images, at times even better than Dickinson’s poems. However, perhaps unsurprisingly, on average, the original poetry scored higher than the machine-made poems. This presents the interesting challenge of automatically creating poetry on par with Emily Dickinson’s.

Preliminary examination comparing poems generated using the custom Markov Chains with those made with a built in Python markovchain package, suggests that the custom model yields better results. Control over stylistic nuances, through, for example, saving words along with their punctuation, seems to help capture Dickinson’s style, and may be relevant to poetic style imitation of other poets. Future studies will need to formally evaluate the performance of the two models, as well as compare to other techniques, such as machine learning approaches.

References


Lamb, C.; Brown, D. G.; and Clarke, C. L. 2015. Human competence in creativity evaluation.


Automated Music Generation for Visual Art through Emotion

Xiaodong Tan
Independent Researcher
frances.tanx@gmail.com

Mathis Antony
CognAI, Hong Kong
mathis.antony@gmail.com

Abstract
We explore methods of generating music from images using emotion as the connection between the visual and auditory domains. Our primary goal is to express visual art in the auditory domain. The resulting music can enrich visual art or provide a form of translation to facilitate enjoying visual art without reliance on the visual system. We use pre-trained image representations and explore two different types of music modelling methods based on RNN and Transformer architectures to build models capable of generating music given an image as input. To evaluate the performance of these methods, preliminary human and machine evaluation are conducted. The results suggest that both music generators are able to express music with an emotional connection.

Introduction
Given a computational method to translate or supplement art in other domains, we could make art accessible to a wider audience and in general enrich the experience of consuming art. Artists express themselves in various domains and a single piece of art may exist in one or more of these domains. Artists can be inspired by art or styles in one domain and create in another domain. The audience can link the art works in different domains (Ranjan, Gabora, and O’Connor 2013) and enjoy the artistic ideas in one domain by being exposed in another domain. These connections might be established through synesthesia, in which perception with one sense is perceived through other senses simultaneously (Ramachandran 2003). It can also be argued that artistic styles exist across domains (Hasenfus 1978) and that art works in two domains can trigger the same perception. While the process of generating and consuming art involves potentially all senses, individual works of art often only exist in one or a few domains (painting, music, ...). Can we use modern computational methods to express art across domains? Different cross-domain art has been studied such as image to music, sculpture to painting, body movement to sound (Marković and Malešević 2018), image to 3-D printable vase (Horn et al. 2015).

In this paper, we focus on the cross-domain image and music translation. Some studies have been conducted on the mapping between visual features (such as color) and acoustic features (such as pitch) (Xiaoying Wu and Ze-Nian Li 2008; Sergio et al. 2015) while others explored existing linked data, such as video and background musics, and used supervised classification to learn their connections (Martín-Gómez et al. 2019). These ideas have been used in applications such as suggesting background sound (Solèr et al. 2016) or descriptive music (Martín-Gómez et al. 2019).

Motivated by the observation that both music and paintings may provoke an emotional response, we build the connection between the two worlds around emotions and use emotions to create in the new domain. (Verma, Dhekane, and Guha 2019) studied the affective relationship between the signals from the two domains. However, this links are not used in generation directly. (Sergio et al. 2015) proposed a mapping of image features to sound features that can potentially trigger the same emotional response from the audience. (Madhok, Goel, and Garg 2018) generates music based on the sentiment detected in people’s facial expression. However, the visual information is not integrated into the model.

The rapid development in AI and deep learning technologies provide new opportunities for multi-modal creation. Transfer-learning and deep image representations are already commonly used in computer vision tasks. These representations are usually pre-trained with models based on CNN architectures on large-scale image datasets, and can be fine-tuned on downstream tasks such as image classification. Cross-domain generative tasks, such as image captioning make good use of these ideas to infuse different signals into generative models (Vinyals et al. 2017). Our methods are greatly inspired by these ideas.

The purpose of this paper is to fill in the gaps in the image-music generation areas with new technologies. The major contribution of the paper is as follows.

1. We propose the task of generating emotionally connected music from images.
2. We use deep learning methods to fuse both the information contained in the image and music into the music generator.
3. We explore the performance of different music generators in terms of how well they convey the emotions in images in their generated music.

The proposed cross-domain generation has applications
as a creative tool, and can also help people with different perceptional preferences to enjoy art from different perceptions. In particular, this kind of system may give visually-impaired people another way to enjoy visual art.

Results and media related to this work are available via a git repository at https://github.com/sudongtan/synesthesia.

Methodology

From a given image and piece of music labeled the same emotion, we extract image and audio features. Then we fuse the two sources of data into a music generator for music generation. An overview of the architecture is shown in figure 1.

![Image to music generator architecture](image)

**Figure 1:** Image to music generator architecture.

Image Feature Extraction

An ImageNet (Deng et al. 2009) pre-trained ResNet (He et al. 2016) CNN model is used to extract images feature representations. In practice we use the ResNet-18 model from torchvision (Paszke et al. 2019; Marcel and Rodriguez 2010) and extract as features the output of the last pooling layer.

Music Modelling

Throughout this project, music is captured in the format of MIDI. Event-based feature extraction is used to capture the timing and dynamics of MIDI files (Simon and Oore 2017), including if there is note or not at a certain timestamp (if it is a note-on or note-off event). For the note-on event, it also included how hard the note is played, i.e., the velocity information.

Multi-modal Encoder-decoder

After image information and music information are encoded separately, they are fused into music decoders, based on the model architecture of the sequence decoder. This method is inspired by other multi-modal generative tasks, such as image captioning (Vinyals et al. 2017) where the image and language information are fused to text decoder and generate text that is related to the image.

Model 1: RNN Decoder

The decoder is composed of a multi-layer RNN architecture. The initial hidden states of the RNN are initialized to vectors that represent the image information. For this the image features are transformed via a matrix multiplication such that they are compatible with the size of the hidden state. The initial input to the RNN is the music representation. By doing this, the decoder has both the image and music information.

In particular, this model takes the velocity information into consideration, so the generated music focus on the timing and dynamics of the music (Simon and Oore 2017).

Model 2: Transformer Decoder

Here a Transformer architecture is used as decoder. Transformers are a self-attention based sequence models. Here we also initialize the initial hidden state of the decoder based on the features extracted from the images as inspired by Zhu et al. (2018).

The Transformer architecture is better than the RNN at capturing the long-term structure of the music (Huang et al. 2018), i.e., the coherence of the music.

Integration of Emotion into the Model

Each training sample consists of a pair of an image and a MIDI file with the same emotion label. In this way, the image information, the music information, as well as their connection are presented to the model. During inference, the image is fed into the model and the music encoding input to the decoder to the is randomly initialized.

Experiments and Results

Training Dataset

**Image** We use a dataset built for emotion recognition (You et al. 2016) as the source of our images. The images are classified into eight types of emotions, namely amusement, anger, awe, contentment, disgust, excitement, fear and sadness.

**Music in MIDI** The MIDI files with emotion labels in the Multi-modal MIREX-like emotion dataset (Malheiro et al. 2013) are used as music data. Originally the dataset partitioned according to emotions into five clusters, which are 1: passionate, rousing, ..., 2: cheerful, good nature, ..., 3: poignant, wistful, ..., 4: humorous, campy, ..., 5: aggressive, tense/anxious, ...

**Pairing Image and Music** We think that the best possible mapping between the labels in the two datasets above is to map music clusters 1 - 5 to image emotion labels excitement, contentment, sadness, amusement and anger respectively. In total there are 17,349 images and 196 midi files.

Model Training and Music Generation

**Model Training** Both models were trained using stochastic gradient descent with Adam optimizer until the validation loss stopped improving.

**Music Generation** During inference, around 600 art photos of emotions amusement, excitement, contentment, sadness and anger from (Machajdik and Hanbury 2010) are used to generate music. The music generated is around 5 - 15 seconds for the RNN model and 10 - 20 seconds for the Transformer model. As expected, the music generated by the RNN model sounds more dynamic but less coherent.
The music generated by the Transformer model is more coherent but also lacks variation.

Evaluation

For evaluations we focus on if an image and a piece of music generated from it invoke similar emotions. To simplify the task of labeling, we consider only their overall positive (contentment, amusement, excitement) or negative (sad, angry) aspect.

Human Evaluation Evaluators are six human beings. The data to be evaluated are 14 pieces of generated music, including 8 from the RNN generator and 6 from the Transformer generator, together with the 14 images from the (Machajdik and Hanbury 2010) dataset from which the pieces of music are generated. Half of the images from each model are labeled positive, the other half are negative. The pieces of music are selected at random but those that were less than 6 seconds long, very repetitive or repeating the training data were discarded. The evaluators are asked to evaluate the emotion of the music and image on a scale from 1 (very negative) to 10 (very positive) without knowing which piece of music is related to which image. To remove subjective biases the absolute ratings are transformed to relative rankings for each participant and medium (Yannakakis, Cowie, and Busso 2017). The results are summarized in figure 2.

![Figure 2: Human evaluation results. Each point denotes the mean relative emotion rating rank for a given image, music pair. The error bars denote the standard deviation. The dashed line is $y = x$, the ideal response.](image)

The correlation coefficient between two mediums is 0.49 indicating that on average the emotion ratings of the generated music is correlated with the emotion rating of the source image. We note there is one obvious outlier towards the bottom right corner where there is consensus that the image conjures positive emotion while the corresponding music conjures negative emotions. We expect that it would take a significantly larger evaluation experiment to determine if there is a significant difference in performance between the two music generators. We also observed that there was better agreement on the images compared to the music among evaluators. In particular there are four pieces of music where the ratings were very dissimilar.

A danger with this evaluation method is that evaluators may express their fondness for a piece of work, instead of the emotion it triggers. For example, a sad but well liked song may be given a high rating instead of the expected low rating in accordance with the triggered negative emotion. As a result, the quality of the generated music may strongly influence the evaluation because songs of good quality are more enjoyable.

Machine Evaluation Here we propose an automated evaluation method and briefly mention a preliminary result for the RNN generator. In order to evaluate the models on a larger scale, we propose to train an emotional correspondence classifier. This classifier takes the image and music features as input and predicts if they express the same emotion (positive label) or not (negative label).

To construct a training sample we choose an image from the image dataset and select a crop from a randomly chosen piece of music labeled with the same emotion (different) for positive (negative) samples. The cropping is done to artificially inflate the number of samples due to the lack of available data. During training we construct positive and negative samples at random and feed them to the classifier. Training this kind of classifier has proven more difficult than anticipated. Preliminary experiments suggest that it is hard to beat 60% validation accuracy.

To evaluate the RNN music generation model we generate 10 pieces of music for 600 images and feed these 6000 pairs into the classifiers. We evaluated this data with one of the better performing classifiers. It predicted 63% of the samples have similar emotions. Therefore we think that even though this seems to be a difficult problem and there is much room for improvement, this method does have potential for automated evaluation of our music generators.

Discussions and Future Work

Technically, one major issue is how to label the emotion of the image and music. Manual browsing of the images and music revealed that some of the labels surprised us. Ideally we would use a stronger connection between the auditory and visual domain and not rely solely on a single word emotion label. Another direction we would like to explore is how to fully use the supervised information. The emotion recognition for image and music respectively can be used to learn better representations of the features.

We explored training a image-music emotion correspondence classifier in the evaluation. We like the fully automated nature and scalability of this evaluation method compared to using human evaluators. It could be incorporated into our workflow of building better music generators. In particular, it could be further explored to play a discriminator role in a Generative Neural Network.

While we would not expect all the music generated with our methods may be suitable for use, we think it provides artists with a exploratory tool to enrich their visual artwork. In practice an artist may generate multiple pieces of music and filter or further process them to their liking. In other
words it could reduce the minimal effort required to find suitable music to supplement a visual art piece from music composition to music curation. We hope this could lead to better accessibility of visual art in the future.

Acknowledgments
We thank the reviewers for valuable criticism and suggestions. In particular for pointing us towards the literature on the ordinal nature of emotions.

References
Ranjan, A.; Gabora, L.; and O’Connor, B. 2013. The cross-domain re-interpretation of artistic ideas.
Melody Similarity and Tempo Diversity as Evolutionary Factors for Music Variations by Genetic Algorithms

Fernando H. Calderon Alvarado  
Social Networks and Human Centered Computing  
National Tsing Hua University  
Hsinchu City, 30013 Taiwan  
fhcalderon87@gmail.com

Wan-Hsuan Lee  
Institute of Information Systems and Applications  
National Tsing Hua University  
Hsinchu City, 30013 Taiwan  
148711499@gmail.com

Yen-Hao Huang  
Institute of Information Systems and Applications  
National Tsing Hua University  
Hsinchu City, 30013 Taiwan  
yenha0218@gmail.com

Yi-Shin Chen  
Institute of Information Systems and Applications  
National Tsing Hua University  
Hsinchu City, 30013 Taiwan  
yishin@gmail.com

Abstract

Music variations consists of modifying an original music piece so that it remains harmonious and provides a sense of novelty while still remain relatable to the original song. The melody should progress in a harmonious way and under a reasonable chord hierarchy, due to this constraints, music variations can be considered as a rule based composition system. In this research, we develop a genetic algorithm to compose a variation on a given music piece. To achieve this we design a novel evaluation function and evolutionary operators which favor the modification process. These are designed to include components of melody similarity and tempo diversity in addition to harmony. The experimental results show that the proposed system can generate variations that preserve musicality. A human evaluation study is also included to validate the proposed evaluation function. Additionally a link to listen to our generated compositions is provided.

Introduction

Music composition has traditionally been another mean of human expression. This process involves a combination of concepts from diverse domains including mathematics, music theory and creativity. Developments in the field of computer music have seen growing efforts in replicating such procedure. Automatic music composition is the process of generating music with the least amount of human intervention. In practice, it can be used to evaluate the degree to which computers can execute the task or as a tool to enhance the human process. This has led to a variety of implementations of music composition systems. Among this efforts we find music variations which are important to study due to their nature of creating over an existing structure. This behavior can provide insights on how the creative process can leverage on prior information and perhaps give us clues on how to apply these concepts in other fields.

The automatic music generation problem is complex and challenging. Recent studies propose music generation systems using neural networks and deep learning techniques and have achieved considerable success (Sturm et al. 2016; Chu, Urtasun, and Fidler 2016; Brunner et al. 2017; Yang, Chou, and Yang 2017; Yu et al. 2017). Another paradigm within the field focuses on composing music in a specific style instead of composing from scratch. Previous studies have focused on the generation of Chinese folk music, Jazz solos and fusions of Flamenco with Argentine Tango (Biles and others; Liu and Ting 2017; Zheng et al. 2017; Luo et al. 2020). Moreover, algorithms have been able to generate specific style harmony based on a given melody. Providing a sense that composition systems can build on top of existing musical structures. Most of these techniques have focused exclusively on harmony instead of a global musical structure as its driving force.

The mentioned studies rely on a target style to generate or adjust a composition. Music variation refers to the process of modifying a musical piece without a target style. The lack of data with original and variation pairs limits the training and modeling by neural networks. Rule-based genetic algorithms do not require training data and have previously achieved good results in music composition and variation (Özcan and Ergal 2007; Majumder and Smith 2018; Alfonseca, Cebrián Ramos, and Ortega 2006). Music variation needs to consider other aspects as compared to traditional composition. In particular, they have to pay attention to the original piece and carefully select modifications that can build on it. This progression must still remain pleasing to a human audience. This requires balancing the amount of diversity added to the variations so that they remain novel while the original songs are still perceivable. In most of the previous studies the evaluation functions driving the evolutionary process rely solely on harmony rules or music theory. We propose a more complete function which considers the following aspects:

- Harmony: Evaluates if the result follows the classical music theory which favor musicality.
- Diversity: Measures the amount of rhythm variation added to the original song while being musically pleasing.
- Similarity: Determines how close is the modified melody to that of the original song.

These measurements can direct the evolutionary process and also be used as evaluation functions for any given pair of music compositions. In conclusion, the contribution of this research is to provide an automatic music variation system which relies on a new evaluation method which considers if the result is harmonious as well as the novelty factor in the
composition while still remaining identifiable to its source.

Methodology

Overview

The proposed music variation system is also similar to the evolutionary process. The modification with higher fitness score has higher potential to survive during the selection. Pitch\(^1\), Interval\(^2\) and Duration\(^3\) are the modifiable features and the basic musical components when composing melody. These features are properly represented in MIDI format which is adopted in this study. Different operators are designed to manipulate them and generate variations while the fitness function based on harmony, similarity and diversity directs the evolutionary process. Evolution is an iterative process, after initialization mutations will be performed on the surviving population until satisfying certain fitness criteria.

Genetic Operators

Initialization Genetic algorithm begins its process with the creation of an initial population. Under this scenario, new note sequences that can be composition variations, need to be created. In music variation, a music section is given hence initial pitch and rhythm are already defined. A new note should not be randomly initialized from a huge search space, adding note series or rearranging notes based on the existing music section is more suitable. 1000 new MIDI sequence with modifications on the given music section are created and those with the highest fitness score are selected. Below we describe and illustrate three novel approaches to initialize these new note sequences.

- **Split note**: This modification splits one note into two of the same pitch with half its original duration. This results in a change in the rhythm while maintaining the same melody contour within the bar (Figure 1).

  ![Figure 1: Split note example.](image1)

- **Exchange notes**: Once the bar is chosen, two selected notes inside the bar will be swapped. The rearrangement of the notes results in a slight variation of the melody (Figure 2).

  ![Figure 2: Exchange notes example.](image2)

- **Add note sequence**: In order to create a different rhythm, melody and more significant variations, one note is broken into a four related chord degree note sequence (Figure 3).

  ![Figure 3: Add note sequence example.](image3)

  \(^1\)The basic component in music and can be regarded as notes which represent the European standard system of 12 equally distributed semitones.

  \(^2\)The distance between two consecutive notes. Consonant intervals, which sound pleasant during the hearing and dissonant intervals, which create a feeling of tension when hearing.

  \(^3\)The length and the timing which one note should occur and finish. The duration of every note in a melody defines the “rhythm”.

Crossover Also called recombination, is a genetic operator to combine the information from two candidates. It is a way to generate new solutions from an existing population. There are multiple ways to implement it such as single point crossover, k-point crossover and uniform crossover. As the given music section is too short, performing k-point crossover will not be as beneficial as single point crossover which will favor keeping the better individual notes from good phrases. This process is illustrated by Figure 4.

  ![Figure 4: Example of crossover for a given phrase.](image4)

Mutation This operator seeks to help the population become better. The mutations occur based on a probability and the positions where they occur are randomly decided. Three mutation alternatives were implemented.

- **Pitch Mutation**: A note out of harmony will be selected and changed to one of a harmony degree based on the previous note.

- **Duration Mutation**: A note is randomly selected and its duration is either doubled or reduced to half and. This enables changes in rhythm while keeping the origin melody.

Fitness Rules

Designing the fitness function in GA can be regarded as a critical point since it determines the quality of evolution. Melody similarity, rhythm diversity and harmony are proposed as components of the fitness function. The total fitness score will result from the sum of these three aspects.
Melody Similarity  Chord analysis can assist in determining which chords and note series are present in a sequence hence enabling a comparison between two sequences. The Spiral Array Model (Chew and Chen 2005) is adopted to obtain the chord in each bar. Pitches are projected into a 3-dimensional space and every collection of notes is represented by a center of effect(CE), which is a point in the interior of the Spiral Array that is the convex combination of the pitch positions weighted by their respective duration. Consequently, the CE of a bar represents its chord. The score for the distance between the original and generated music pieces can be obtained by equation (1), were b is the total number of bars in the song.

\[
S_c = 100 - \frac{\sum_{i=0}^{n}(CE_i - CE'_i)}{b} \tag{1}
\]

Tempo Diversity  To evaluate the differences in tempo we adopt the metrical complexity (Thul and Toussaint 2008). This measurement uses metricity(W), the sum of all the metrical accents of the beats present in a rhythm, to obtain a tempo complexity complexity. Equations 2 and 3 present the tempo diversity score calculation where \( maxW_i \) is the maximum metricity for 5 beats and \( W_i \) is the actual metricity for every 5 beats in the generated song.

\[
Tempo\text{Complexity} = \sum_{i=1}^{n} maxW_i - \sum_{i=1}^{n} W_i \tag{2}
\]

\[
S_d = \sum_{i=0}^{n} Tempo\text{Complexity}_i \tag{3}
\]

Harmony  For measuring the harmony of a song, the fitness function evaluates every sequence according to music theory. It examines every note from the sequence, whenever a rule is matched, the fitness score is modified accordingly. The higher score means the sequence violates less rules. These rules mainly focus on the basic consonance between consecutive notes and the harmony note with chord sequence, the score arrangement of each rule is listed in Table 1 and its calculation denoted by equation (4).

\[
S_h = \sum_{i=0}^{n}(Rule_i * d_i)/n \tag{4}
\]

Experiments

Measurement Validation

First, to verify the designed harmony rules can scale the song correctly or not, six testing songs including three from major tonal and three from the minor tonal are evaluated. The harmony score is calculated to see if the proposed rules can scale the harmony degree accordingly. The results presented in Table 2 support theoretical agreements stating harmony is dependent on Major tonality.

<table>
<thead>
<tr>
<th>No</th>
<th>Rule</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2 consecutive notes are the same (C, C)</td>
<td>+3</td>
</tr>
<tr>
<td>2</td>
<td>2 consecutive notes are Major 2nd (C, D)</td>
<td>+3</td>
</tr>
<tr>
<td>3</td>
<td>2 consecutive notes are Major 3rd (C, E)</td>
<td>+3</td>
</tr>
<tr>
<td>4</td>
<td>2 consecutive notes are Perfect fourth (C, F)</td>
<td>+3</td>
</tr>
<tr>
<td>5</td>
<td>2 consecutive notes are Perfect fifth (C, G)</td>
<td>+3</td>
</tr>
<tr>
<td>6</td>
<td>Big jump between notes (degree &gt; 5)</td>
<td>-8</td>
</tr>
<tr>
<td>7</td>
<td>The note is a chord root note</td>
<td>+5</td>
</tr>
<tr>
<td>8</td>
<td>The note is a second chord note</td>
<td>+4</td>
</tr>
<tr>
<td>9</td>
<td>The note is a third chord note</td>
<td>+4</td>
</tr>
<tr>
<td>10</td>
<td>The note is in the scale (C major)</td>
<td>+2</td>
</tr>
</tbody>
</table>

Table 1: Fitness Score for Every Rule

The fitness function was tested on a human composed music variation. “12 variation on Twinkle Twinkle Little Star” consists of improvisations on each section composed by Mozart. Table 3 presents the resulting scores for each varying section. This experiment demonstrates the proposed measurements can properly model the target characteristics pertinent to music variation.

<table>
<thead>
<tr>
<th>No</th>
<th>Song</th>
<th>Tonal</th>
<th>Harmony Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Through the Arbor</td>
<td>Major</td>
<td>106.03</td>
</tr>
<tr>
<td>2</td>
<td>Little Star (Mozart)</td>
<td>Major</td>
<td>99.8</td>
</tr>
<tr>
<td>3</td>
<td>Minuet G (Bach)</td>
<td>Major</td>
<td>100.2</td>
</tr>
<tr>
<td>4</td>
<td>Sonata No. 5</td>
<td>Minor</td>
<td>66</td>
</tr>
<tr>
<td>5</td>
<td>Concerto 5 (Beethoven)</td>
<td>Minor</td>
<td>68</td>
</tr>
<tr>
<td>6</td>
<td>Concerto 23 (Mozart)</td>
<td>Minor</td>
<td>71</td>
</tr>
</tbody>
</table>

Table 2: Harmony Score in Different Type of Music

<table>
<thead>
<tr>
<th>Section</th>
<th>Harmony</th>
<th>Similarity</th>
<th>Diversity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100.88</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>101.33</td>
<td>96.8</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>89.8</td>
<td>83.25</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>89.33</td>
<td>92.5</td>
<td>8.6</td>
</tr>
<tr>
<td>5</td>
<td>87.79</td>
<td>84.88</td>
<td>8.2</td>
</tr>
</tbody>
</table>

Table 3: Harmony and Similarity Score for Little Star Composed by Mozart.

Music Variation Results

The proposed system was tested with songs from multiple genres ranging from classical music to pop. Parameters for the GA were set at 8 bar sequence length, population size of 100, 0.5 crossover and 0.05 mutation running for 600 generations. A demo of the results is available at “https://sharon1018.github.io/”. To evaluate the impact of each of the proposed measurements on the fitness function three different variations are presented for each song, incrementally adding an additional component. Samples of the music scores for original an variation songs are presented in Figures 5 and 6.

Through visual inspection and listening to the demo it can be perceived how the musicality is preserved. It is also a
Human Evaluation of the Variation Results

To rate the quality of the generated songs human evaluation was performed. A total of 20 test subjects, university students aged 22 to 26 with no specific music background, scored the variation results for the different combinations of generated songs as presented in the demo. Figure 7 shows the average harmony score rating, all samples score above 6/10 indicating the songs preserve and even improve their harmonious musical quality while integrating the new features. The samples were also rated in terms of similarity and diversity displaying similar results.

Figure 7: Harmony score for different measurements.

Conclusion

The automated evolutionary approach for music variation is discussed and evaluated in this research. The proposed operators and fitness function guide the evolution process and are able to generate coherent music pieces. The system successfully generated samples that achieve the desired surprise factor in variations and still being relatable to the original songs.

References


ERwEM: Events Represented with Emotive Music Using Topic-Filtered Tweets

Makayla Harris, Hunter Harris, Paul Bodily
Computer Science Department
Idaho State University
Pocatello, ID 83209 USA
{thommaka, harrhunt, bodipaul}@isu.edu

Abstract

We present ERwEM (Events Represented with Emotive Music), a system that generates cross-domain artefacts between text and music by drawing from emotional sentiments. Our system uses keywords extracted from a news article to filter live-stream tweets containing these keywords and analyze the tweet collection’s emotion data. It then uses the emotion sentiment (a scale from positive to negative) to influence the mood of the music it writes using an adapted bi-axial long short term memory (BALSTM) model. From this, our system determines which mode the generated song will be transposed to. We use the FACE model to characterize the creative aspects of our system as a collection of generative acts. We provide examples of the artefacts our system generates as well as the response we received from external sources who reviewed the generated songs.

Introduction

Those in the field of computational creativity are consistently exploring new domains to apply CC and methods by which systems can produce creative artefacts. One such domain that continues to grow is music composition. In this domain, songwriters often draw on their emotional sentiment towards a concept or event to write music that reflects the way they feel about it. This is often reflected in the lyrics produced or the song’s mode, which is the combination of its scale and tonal center. Our system explores this experience based method of songwriting by writing emotive songs that represent events or topics found in the news. Our system uses the these news articles and emotion values extracted from tweets to express emotion in its songs. This method for music creation led us to name our system ERwEM (Events Represented with Emotive Music).

In this paper, we leverage the correlation between modality and emotion to allow our system to generate emotive music. There is a strong correlation between the speed and modality of a song and the emotion that is attributed to the song (Fritz et al. 2009). This was found specifically for western style music and modes which are the same modes we employ in our system.

Related Works

There have been many forms of music generation ranging from statistical models (Conklin 2003) and Markov models (Schulze and Van Der Merwe 2010), to neural networks (Yang, Chou, and Yang 2017). There have also been studies on making emotive music from text, including the study on TransProse (Davis and Mohammad 2015). This study uses emotional data (a numerical measurement of the emotions found in text using the NRC Word-Emotion Association Lexicon) from literature to create music using parameters for pitch, tempo, and key; however, the music lacks a consistent vertical structure because it uses two independently written melodic lines played simultaneously. We intend to use this same emotional data to transform songs generated by a model able to create vertical structure in a song.

A study comparing the differences between neural networks and Markov model music generation (Cruz 2019), shows that neural networks generate music that more closely matches its training set than Markov models. A paper by Johnson describes one such neural network: a biaxial long short-term memory network (BALSTM) (Johnson 2017). There are other long short-term memory networks (LSTM) used for music generation (Sturm et al. 2019; Eck and Schmidhuber 2002; Boulanger-Lewandowsk et al. 2012), but the BALSTM is efficient in encoding both temporal and pitch patterns from music. Transposition invariance is a unique attribute of the BALSTM which allows training and generation in any key. It also allows for it to sample from a dataset containing songs with varying keys. Since MIDI files are often missing key information, this makes the key invariant model more convenient to use. Otherwise, key identification and transposition to a common key would be necessary for each song in the training data. We adapt Johnson’s BALSTM as the music generation component of ERwEM.

The Nottingham Database¹ is a set of 1200 American and British folk songs. The MIDI files that make up this dataset are simple songs that consist of a melody played over chords. We found this dataset to generate song-like results more consistently and with fewer training iterations than other datasets we used. As a result, we use the parame-

---
¹https://ifdo.ca/~seymour/nottingham/nottingham.html
ers created from using this training dataset.

To extract emotion from text, we use the NRC Word-Emotion Association Lexicon (Mohammad and Turney 2013). Using this lexicon and Twitter, our system finds inspiration for the emotions it will use in its generation process.

There are several papers on how creativity can be measured and attributed to a system such as Ritchie’s metrics (Ritchie 2007), and Colton’s FACE model (Colton, Charney, and Pease 2011) which we relate to our system. Ventura also describes a general structure for CC systems (Ventura 2017) as well as what it means for a system to be merely generative (Ventura 2016). He discusses the spectrum of stages for a creative system that we use to show where our system is located in the spectrum of creativity and what is necessary for higher levels of creativity.

**Methods**

In this section we describe the design and construction of ERwEM. We first give an overview of our system and the task it is designed to accomplish. We then provide a detailed description of the components that make up our system and the processes they use to reach our goal.

**ERwEM System Overview**

ERwEM is designed to generate emotive music from events it finds from The Guardian news outlet. Our system begins by selecting a news article and extracting the keywords from it. Using those keywords, the system filters Twitter for live tweets that contain any of the keywords gathered from the news article. Emotions are gathered from the collected tweets using the NRC Word-Association Lexicon and mapped to a musical mode. From there, our system generates a new song using a BALSTM and transposes the song to the aforementioned mode. We trained the BALSTM on the Nottingham Database for its simplicity and consistent structure which generated the most songlike pieces. The complete artefact generated by the system consists of the final composition and its framing (see Figure 1).

To show the components of our system that contribute to its creativity, we analyze it using the FACE model. The FACE model defines the creative components of a system as a tuple of generative acts. We argue the tuple representing ERwEM to be \( < F^9, C^9, E^9 > \). These generative acts are briefly outlined below.

- **\( F^9 \):** Framing generated from news, tweets, and emotions
- **\( C^9 \):** Concepts created from extracted keywords and their perceived emotion
- **\( E^9 \):** Expression developed by transposing a generated song to a mode determined by emotion

**Concept** ERwEM generates concepts for its creative process from news and tweets. It begins by selecting a news article \( a \in A \) where \( A \) is the set of all articles available in The Guardian’s API. From \( a \) we extract \( n \) number of keywords \( k \) where the set of keywords for a given article is \( a_k \). We arbitrarily chose \( n = 3 \) in our system. We extract the keywords from the article using a chi-square test for independence to find the words that are statistically significant. We use approximately twenty years worth of articles from The Guardian news outlet to build the statistical model. Using those keywords \( a_k \) we filter live tweets \( t \) from Twitter giving the set \( a_k \), (see Figure 2). We obtain the emotions \( \varepsilon \) from \( a_k \) by using the NRC Word-Emotion Association Lexicon where \( \varepsilon \) is a set of emotion and emotion score pairs (see Table 1). This process gives us the concept \( c^9 = (a, a_k, e_k, \varepsilon) \).

**Expression** The expression, or artefact, is generated by using a concept \( c^9 \). The system begins by composing a song \( s \) in the form of a MIDI file. Then, using the emotion data \( \varepsilon \) from the \( c^9 \), we calculate the overall brightness \( b \) of the expression as:

\[
b = \frac{\varepsilon_{pos} - \varepsilon_{neg}}{\varepsilon_{pos} + \varepsilon_{neg}}
\]

Where \( \varepsilon_{pos} \) is the count of words with the positive attribute marked and \( \varepsilon_{neg} \) is the count of words with the negative attribute marked. This gives \( b \) a range of \([-1, 1]\). Our system determines the mode \( m \) for the expression by mapping the set of the seven common modes \( M \) to the range \([-1, 1]\). This is accomplished by spacing the seven modes equally across the range such that the brightest mode is equal to 1 and the darkest mode is equal to -1 (see Figure 3). The order of these modes reflects the order described in music theory. Locrian is the darkest mode and each mode is subsequently brighter than the last such that Lydian is the brightest.
Table 1: An example of the emotional data extracted from the tweet pictured above. This was accomplished using the NRC Word-Emotion Association Lexicon which totals the emotion score for each word in the tweet. The relative distribution of these is represented in parantheses.

<table>
<thead>
<tr>
<th>Emotion</th>
<th>Value</th>
<th>Emotion</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>1 (20%)</td>
<td>Negative</td>
<td>4 (80%)</td>
</tr>
<tr>
<td>Anger</td>
<td>0 (0%)</td>
<td>Anticipation</td>
<td>0 (0%)</td>
</tr>
<tr>
<td>Disgust</td>
<td>0 (0%)</td>
<td>Fear</td>
<td>2 (50%)</td>
</tr>
<tr>
<td>Joy</td>
<td>0 (0%)</td>
<td>Sadness</td>
<td>2 (50%)</td>
</tr>
<tr>
<td>Surprise</td>
<td>0 (0%)</td>
<td>Trust</td>
<td>0 (0%)</td>
</tr>
</tbody>
</table>

Figure 3: The mapping of the seven common modes to the range \([-1,1]\] where the relative brightness of the mode increases from left to right (e.g., locrian=darkest, lydian=brightest).

We will denote the current mode of our song as \(s_{m_0}\) and the mode it is to be transposed to as \(s_m\). To programmatically transpose \(s_{m_0} \rightarrow s_m\), we created a series of vectors that will act as transformations from one mode to the next closest mode in descending order (see Table 2). Using this model, we are able to construct a transformation vector from any one mode to any other mode. As an example, we show the process for creating a transformation vector from ionian (major) to aeolian (minor). Consider the following equation.

\[
\begin{bmatrix}
    0 \\
    0 \\
    0 \\
    -1 \\
    0 \\
    0 \\
    0
\end{bmatrix} \rightarrow
\begin{bmatrix}
    0 \\
    0 \\
    0 \\
    0 \\
    -1 \\
    -1 \\
    0
\end{bmatrix}
\]

By adding the transformation vectors for each successive transposition from ionian to aeolian, we are left with a new transformation vector that will take any song in ionian and transpose it to aeolian (this process of transposing maintains the current key the song is in and only affects the mode).

An example of this on the key of C in mode ionian is shown below.

\[
\begin{array}{cccc}
C \text{ Ionian} & I_{Io}^{C \rightarrow Ae} & C \text{ Aeolian} \\
0 & 0 & 0 \\
2 & 0 & 2 \\
4 & -1 & 3 \\
5 & 0 & 5 \\
7 & 0 & 7 \\
9 & -1 & 8 \\
11 & -1 & 10 \\
\end{array}
\]

This process gives us our resulting expression \(e^g = (m, s_m)\) where \(s_m\) is the transposed version of the originally generated song \(s\). To tie our concept and expression together, our system proceeds to construct framing for the expression from its associated concept.

**Framing** Our system creates framing \(f^g\) for \(e^g\) by using the information in \(e^g\). We create the title for the current expression \(e^g_{\text{name}}\) by joining the three keywords \(a_k\) with underscores, and we choose the top two emotions from \(\varepsilon\) (excluding positive and negative) to be \(\varepsilon_1\) and \(\varepsilon_2\). By using the article title \(a_{\text{title}}\), the keywords \(a_k\), the top two emotions \(\varepsilon_1\) and \(\varepsilon_2\), and the mode \(m\), our system constructs the framing using the following template.

While working on my project \(e^g_{\text{name}}\), I found a news article titled: "\(a_{\text{title}}\)". I checked Twitter to see what other people had to say relating to the topics \(a_k\) that I found in the article. What I read from the tweets made me feel \(\varepsilon_1\) and \(\varepsilon_2\). As a result, I decided to write a song in \(m\).

This framing \(f^g\) is displayed alongside the \(e^g\) to enhance the creativity of our system.
Results

In this section, we will describe the results from building and testing our system as well as some of the ways we evaluated its artefacts.

ERwEM describes the context from which it gathered its inspiration as well as the decision it made from the context. This helps our system explain why and how it generated the songs that it does and enhances its creativity by acting as the $F^9$ generative act. This helps people who listen to the songs connect with the process by which ERwEM created its artefacts and understand the inspiration behind them. We believe this can improve the perceived creativity of our system.

A snippet of a song generated by our system can be seen notated musically in Figure 4. Audio examples of the generated artefacts can be listened to on ERwEM’s SoundCloud2. The resulting artefacts had a moderate level of variance, generating songs in the following modes: aeolian, dorian, mixolydian, and ionian. We are led to believe the reason our system has not written in the darkest and brightest modes is due to the rarity of finding extreme emotion values from a large sample of tweets. We have determined this not to be a flaw in our system because songs are rarely written by human composers in the darkest and brightest modes.

Conclusion

In our efforts to create a system that can express emotive music, we attempt to simulate the way songwriters incorporate personal and perceived experiences about concepts through our system’s use of emotional analysis applied to music. Our system is currently in between Generalization and Filtration in Ventura’s spectrum of generative systems. Our system could be attributed a higher level of creativity if we implement a fitness evaluation function for the works produced by our system. This is a candidate for future work with ERwEM.

Although our system struggles in creating songs that sound vastly different from each other, we argue this is something that human musicians also struggle to do. We recognize the limited dataset and training time to be limitations of our system and propose more advanced methods of training and generation for future work. We also note the BALSTM model creates music with vertical structure, but does not have horizontal structure. We recognise this as a limitation of our system and would explore different models of music generation that have greater capabilities in creating horizontal structure throughout songs. We would also use the emotion data to influence tempo, instrumentation, pitch, etc., in its song generation.

References

Boulanger-Lewandowsk, N.; YoshuaBengio, P. V.; Gray, P.; and Naguri, C. 2012. Modeling temporal dependencies in high-dimensional sequences. ICML.


Citezee.


2https://soundcloud.com/anonymized
Creating Latent Spaces for Modern Music Genre Rhythms
Using Minimal Training Data

Gabriel Vigliensoni, Louis McCallum, and Rebecca Fiebrink
1Department of Computing, Goldsmiths University of London, UK
2Creative Computing Institute, University of the Arts London, UK
g.vigliensoni@gold.ac.uk

Abstract
In this paper we present R-VAE, a system designed for the exploration of latent spaces of musical rhythms. Unlike most previous work in rhythm modeling, R-VAE can be trained with small datasets, enabling rapid customization and exploration by individual users. R-VAE employs a data representation that encodes simple and compound meter rhythms. To the best of our knowledge, this is the first time that a network architecture has been used to encode rhythms with these characteristics, which are common in some modern popular music genres.

Introduction
In this paper, we present research on customizing a variational autoencoder (VAE) neural network to play with musical rhythms encoded within a latent space. To enable customization and personalization, the network can be trained with as few as one dozen MIDI clips with rhythms.

Additionally, our approach employs a data structure that is capable of encoding rhythms with binary, ternary, or a combined metrical grid. A metrical grid can be explained as the main ratio by which the onsets of notes are placed in a measure. Music where the beats are split in two—a binary grid—is in simple meter. Music where the beats are split in three—a ternary grid—is in compound meter. Many modern music genres, such as footwork, gqom, dembow, or trap, can be characterized by rhythmic elements using a compound meter. In these rhythms, the main meter is usually simple but there are elements that are placed on an overlaid ternary grid. To the best of our knowledge, this work is the first time that a network architecture has been used to encode rhythms that exhibit a combined binary and ternary grid.

Related Work
Several recent projects have aimed to use machine learning to encode the regularities in rhythmic patterns present in user-provided examples into a model, so that this model can afterwards be used to sample rhythms from the original distribution or generate new, unheard rhythms. For instance, Choi, Fazekas, and Sandler (2016) created a model trained on drum patterns from songs by Metallica to generate new rhythmic sequences. Their approach was based on a text-based LSTM (Long Short Term Memory) network, so that they had to adapt and encode the rhythm onsets to fit the data representation. They limited the number of events in a bar to 16 by quantizing the drum tracks to 16th notes in a simple meter binary grid. They used 60 MIDI files of drum tracks as training data and created drum sequences that were “reasonable rock drum patterns.” However, since the data representation they used did not encode the deviations of onsets from the grid—known as microtiming—or how hard the onsets were struck—known as velocity in the MIDI domain—the resulting patterns were very rigid.

Instead of focusing on a specific musical artist or style, Nikolov (2016) trained another LSTM network on drum patterns from a wide range of music genres with the goal of creating a general model for rhythm generation. This work used professionally produced MIDI loops, increased the quantization grid to a 32nd note, but used only rhythms with a 4/4 time signature and simple meter. Nikolov described some of the resulting examples as “musically coherent patterns” but only in a short timescale.

With the goal of learning longer-term musical structure, researchers of the Google Magenta team experimented with language modeling and LSTMs to encode and generate melodies and drum patterns. They released network architectures designed to learn representations of melodies and rhythms encoded in a symbolic format, as well as models pre-trained on large datasets containing “thousands of MIDI files” of undisclosed origin. The data representation did not encode microtimings or velocities.

Using the data representations implemented in the Magenta projects, and with the goal of modeling sequences with even longer term structure, the Magenta team released MusicVAE (Roberts et al. 2018). This network used a hierarchical variational autoencoder (VAE) architecture to encode and generate melodies, drums, and “trios” consisting of a drum part, a bass line, and a melodic line. For these three categories, the Magenta team also released models that were trained on a very large dataset of more than 1.5 million MIDI files collected from the web. Roberts et al. reported that MusicVAE was able to generate 2-bar drum sequences reliably.

1https://github.com/magenta/magenta/tree/master/magenta/models/melody_rnn
2https://github.com/magenta/magenta/tree/master/magenta/models/drums_rnn
but failed when trying to reconstruct 16-bar sequences.

In the aforementioned approaches to drum pattern modeling, only the position of the drum onsets was encoded, not their microtimings or velocity. These two characteristics are important for giving the drum loops a human feel, or groove. In order to overcome these flaws, the Magenta team released GrooVAE (Gillick et al., 2019), a data representation and a set of models trained on real drum performances. Again, this work used quantization to a 16th note grid in both the data structure and models, resulting in an inability to encode rhythms such as footwork or trap, which commonly have rhythmic elements in compound meter. As a result, these elements are quantized incorrectly into the binary grid, probably losing their rhythmic signature or “feel.”

A number of applications have been released based on the Magenta data representation and their pre-trained models. For example, the Neural Drum Machine3 is a web-based application in which the user seeds the system with a short rhythmic sequence, and the model “imagines” the continuation. Beat Blender4 packages MusicVAE pre-trained models in a web application in which the user plays back patterns and creates paths in a latent space filled with rhythms. The Drum Beats Latent Space Explorer5 is another web application that uses a VAE architecture trained on 33K MIDI drum files to learn a bi-dimensional latent space representation that can be explored in a browser. All the rhythms in these applications are quantized to 16th notes and so their data structure is only able to decode rhythms based on a binary grid of simple meter.

### R-VAE

**Motivation**

Our goal is to design a system that can generate a series of models using minimal training data, to better enable artists without extensive computational resources to build and explore bespoke rhythm models. Further, we would like this system to encode the onsets, velocities, and microtimings of rhythms, and to allow the encoding of simple or compound meter rhythms, or their combination. Once a model is trained, performers using the system should be able to explore the latent space of the model and retrieve rhythmic patterns as if they were moving a playback head on it.

Most of the architectures, models, and applications we have reviewed have been trained using very large datasets of rhythms. For example, MusicVAE models were trained on about 1.5 million unique MIDI files (Roberts et al., 2018), and the GrooVAE and Expanded Groove MIDI Dataset models were trained on more than 13 and 444 hours of music, respectively, performed by professional drummers and recorded in both MIDI and audio formats (Gillick et al., 2019; Callender, Hawthorne, and Engel, 2020). The goal of these representations was to learn the groove—the human feel—in drum performances. To achieve this, the authors encoded in a VAE network the onsets, velocities, and microtimings of the drum hits.

These approaches entail practical challenges, both in that they require large-scale datasets and in that they are trying to learn commonalities from disparate data. Creating generic models from large and diverse data can be interesting and reasonable from a computational point of view, but this can also hinder customization. Individual creative practitioners usually do not have access to large datasets or the processing power to train such large models. On the other hand, the ability to train models from smaller datasets can enable the modeling of niche genres or even a personal style, while requiring fewer resources for training.

Some prior work has aimed to empower individual expression and personalization of generative models by enabling training from smaller datasets. Dinculescu, Engel, and Roberts (2019) introduced MidiMe, an approach to quickly train a small model to control a much larger and generic latent variable model. Their system learns a compressed representation of the already encoded latent vectors of MusicVAE and generates musical melodies from only portions of its latent space based on MIDI files with melodies provided by users by means of a web-based app. Other work has enabled musicians to create bespoke supervised learning systems with small training sets; for instance, Wekinator (Fiebrink, Trueman, and Cook, 2009) uses shallow multi-layer perceptron neural networks to learn bespoke mapping functions (e.g., from a performer’s gestures to sound synthesis parameters) using small datasets generated by musicians in realtime.

### Implementation

Autoencoders (Kingma and Welling, 2014) can learn a compact representation of the training data that captures important factors of variation in the dataset. Points in the latent space map to realistic datapoints, and nearby points map to semantically similar (or here, musically similar) examples. Variational autoencoders (VAEs), in particular, assume the training data has an underlying probability distribution and attempt to find the parameters of the distribution. Once those parameters are found, we can sample the space to generate data that will follow the original distribution. In other words, in VAEs the generated data will be related to (but not necessarily the same as) the source data. This make VAEs a good network topology for creating generative models of rhythms.

We have implemented a variational autoencoder-based system, called R-VAE, which, for the first time, encodes simple and compound meter rhythms. It also encodes the onsets, velocities, and microtimings of rhythms, and can be trained using small datasets. We released a web-based app that can be used as a rhythm model player, enabling people to explore rhythmic latent spaces and make music directly in the browser. We based the implementation of our rhythm explorer on the Tensorflow.js VAE implementation called tfjs-vae6 and the M4L.RhythmVAE rhythm generator device (Tokui, 2020). While the former provides the Tensorflow backend for the VAE, the latter provides a data

---

4https://experiments.withgoogle.com/ai/beat-blender/view/
5https://towardsdatascience.com/drum-patterns-from-latent-space-23d59dd9d827
6https://github.com/songer1993/tfjs-vae
structure based on the one by Gillick et al. (2019) that encodes the onsets of rhythms, their velocities, and microtimings. M4L.RhythmVAE also comes conveniently packed as a Node for Max application that can be opened as a Max for Live device in Ableton Live, a popular digital audio workstation.

The configuration for training our model consists of a vanilla VAE architecture with 864 dimensions for the input, 512 for the intermediate layer, and 2 dimensions for the latent space. The batch size is set to 64, the optimization algorithm to Adam, and the activation function to LeakyReLU. The favouring of fully connected feedforward layers by Tokui instead of Gillick et al.’s bidirectional LSTMs allows for faster training using CPUs and we compared the performance of this implementation with much larger and complex architectures such as MusicVAE and GrooVAE. We found it required considerably less data and processing power to converge into a useful model.

**Binary and ternary representations** In R-VAE we extended the internal data representation of M4L.RhythmVAE to encode simple and compound meter rhythms, as well as their combination. Most previous approaches used only sixteen 16th notes in one bar of 4/4 time, corresponding to a resolution of four ticks (i.e., subdivisions) per quarter note. However, the encoding of most modern music genres needs a much finer grid of up to a 32nd triplet note, which we then chose as the basic unit in our data representation. Then, the encoding of one bar of 4/4 time in R-VAE comprises three matrices (for onsets, velocities, and microtimings) of dimensions 96 × 3. These dimensions represent 24 ticks × 4 quarter notes × 3 drum instruments. Although GrooVAE and M4L.RhythmVAE work with nine canonical drum categories, for this project we only work with the three main drum instruments in modern popular music: kick, snare, and hi-hat.

**User interface** The chosen VAE topology projects the input matrices to a two-dimensional space. As can be seen in Fig. 1, this rhythmic latent space is presented as a two-dimensional plane that the practitioner can play using the analogy of controlling a playback head. Clicking on any given point of the latent space will retrieve and decode a rhythm sequence. As expected, the patterns mimic closely those ones in the training data with the additional benefit of being able to interpolate between them by dragging the playback head, or to extrapolate to new ones when moving to a new zone in the space.

The user interface exposes two variables to control and add variability to the network decoding. **Threshold** controls the complexity of the patterns decoded from the latent space, and **noise** rules the precision of the mapping between the performance space to the latent space. Mute buttons per instrument allow the performer to silence instruments at any given moment. These parameters can be seen on Fig. 1 as part of the web-based player application. This browser-based version also features MIDI output so that performers can integrate R-VAE with external standalone devices and software-base sound engines.

A video demonstrating the capabilities of R-VAE and snippets of renditions performed with it can be accessed at https://vimeo.com/422294058. Both implementations of R-VAE, for Ableton Live⁷ and the R-VAE-JS browser-based model player,⁸ are available.

---

⁷https://github.com/vigliensoni/R-VAE
⁸https://github.com/vigliensoni/R-VAE-JS

Figure 1: User interface of R-VAE-JS web application. The latent space can be explored by using a playback head represented by the yellow circle, in the black performance space on the right of the figure. Additional knobs for threshold and noise help the performer to control how to the latent space is sampled. Mute buttons enable the performer to silent individual instruments. User interface elements on the left side provide additional visual feedback.
Discussion

We have investigated the viability of our system by encoding rhythms from music genres such as footwork, gqom, and trap, and using the models in musical improvisation and performance. When experimenting with data selection and model training, we found that the system was able to learn useful and playable models with as low as one dozen MIDI clips. However, when we increased the number of clips to a few dozen, the learned latent spaces were richer and exhibited a more even topology, helping the performer to create smoother interpolations when moving the playback head between rhythms in different zones of the performance space. The use of R-VAE in creative practice demonstrates that its data representation is able to model accurately simple and compound meter rhythms, opening possibilities for using this tool with other rhythms exhibiting these characteristics.

Even if not perfect, individually crafted models from smaller datasets could prove to be useful and inspiring to the creative practitioner. The amount of data needed to generate creatively interesting models will vary with context and intention, but small data may be more suited to generate custom models that are good for exploring a specific idea or creative concept (Fiebrink, Trueman, and Cook 2009).

Interacting and performing with latent spaces encoding rhythms pose excellent questions, from both a computational and musical point of view. For example, how can we characterize the latent space in terms the smoothness of the interpolations? What is a good metric to measure the richness of the encoded space? Musical performance contains some pertinent differences to other fields unaddressed by current technical literature. For example, when measuring rhythmic similarity, the apparently small change of moving a few onsets from a 16th grid to its closest triplets is small in terms of edit distance, but it can have a large perceptual impact. Furthermore, there are some instrumental hierarchies when working with rhythms. For example, changing a few hi-hats in a rhythmic sequence is likely to have a subtler perceptual influence than changing a drum kick pattern.

Conclusions and Future Work

We have presented R-VAE, a system designed for the exploration of latent spaces of simple and compound meter rhythms, a common combination in modern musical genres. The ability to use it with small datasets can enable the modeling of niche genres, while requiring fewer resources for training. We used R-VAE to learn models of a few modern music genre rhythms and used them in creative music production and performance. Some of the insights we learned are: (i) VAEs are capable of encoding compound meter rhythms; (ii) small training data is enough to create a useful and playable rhythmic latent space; but (iii) if the data is too small, the space can be perceived as a discrete collections of zones, instead of a contiguous space.

Research is needed to overcome the issues found and improve the system. For example, visual feedback may help performers to visualize the topology of the latent space, so that they know if they are in zones with specific rhythms or in zones of transition. Along the same lines, and in particular for small datasets, displaying where the original training data points are encoded in the latent space may provide visual guidance to explore the space. As a result, there is a pressing question about how to best incorporate this visual feedback into the system. Additional improvements are useful and needed. The web-based application may benefit from making the training process available directly in the browser, so that there is no need for an additional application. Extra playability and variability of the rhythmic patterns can be achieved by extending the number of bars encoded in the space and the number of drum instruments.

Our experience with R-VAE has reinforced the idea that a system for the exploration of latent spaces of musical rhythms is worth pursuing further. Systems like this could be also used for browsing through libraries of rhythms, common in contemporary music production.

Acknowledgments

This research has been supported by Fonds de recherche du Québec – Société et culture (FRQSC) through a research creation grant (Ref. 2020-B5-272998).

References


Abstract
Cross-Domain creative systems can learn how to creatively map and/or transform semantic concepts from remote domains to their own. This project sets out to explore how a computational system relies on what it has learned from analyzing images to produce creative work. Specifically, the system presented here extracts semantic content and themes from an input image and uses this knowledge to creatively generate emotionally-appropriate musical artifacts to accompany the image. An independent experiment was conducted to provide a performance assessment and direct future work.

Source: github.com/DLasher95/TransEmote

Introduction
A hallmark of creative minds is the ability to apply knowledge from between creative domains. The concept of “cross-domain systems” allows us to take substantive inspiration from one domain and transfer it to another. Our system exists in the musical domain and extracts its inspiration from the space of visual arts. We computationally bridge the gap of these domains because, devoid of words, both music composition and imagery portray vivid and emotional narratives.

Our system, called TransEmote for its ability to translate emotive expressions, analyzes the content of an input photographic image and uses artificial intelligence (AI) to creatively generate emotionally-equivalent music to accompany the image. The system extracts psychology-based emotions, derived from the semantic content and colors present, and forms an emotional profile for the input. This way, TransEmote is able to extract multiple, complicated emotional themes and map them into semantically-appropriate music to accompany the image. There exist many benefits in pursuing this potential line of research:

• An aid to the visually impaired, helping to convey the emotional magnitude of photos, and illustrations.

• A rudimentary approach to developing dynamic soundtracks for large media, such as movies or games.

• A tool for emotional industries, like mental health clinics, to generate calming and/or happy music without having the domain knowledge to compose music themselves.

The challenge in composing music, just as in photography, is the magnitude of choices. We present a novel approach to extract visually-portrayed emotions, as well as a number of mapping rules to generate various elements of music, such as tempo or the major/minor key. Our goal with this project is to present an early framework by which the music and the visual arts can be further connected through creative AI.

Related Works
Cross-domain creative systems are few within the community. Intelligent music composition has seen more thorough treatment. We consider each in turn.

Cross-Domain Systems Text-to-music currently has difficulties with reliably mapping text to changes in sound. For this reason, prior text-to-music systems tend to rely on shallow features of text to direct generation rather than semantic context. Rangarajan (2015), for instance, proposed three methods of mapping text to music.

1. Mapping letters to notes, and frequencies to duration.

2. Mapping vowels to notes and note duration.

3. Mapping vowels and their respective uses to notes.

The Transpose project set out to generate music that captured emotion in literature by studying changes in the distribution of emotional words (Davis and Mohammad 2014). Transpose assigned major keys to novels with more positive emotions and minor keys with more negative emotions. It also connected the frequency of words with the tempo. Their approach laid the groundwork for cross-domain music generation.

Intelligent Music Composition There has been a significant amount of work done to map emotions to discretized parameters, such as tempo indicating happiness (Hunter, Schellenberg, and Schimmack 2010) and melody indicating calmness. The resources concerning emotional music generation is scarce, however.

It is important to note that these parameters are largely common in most cultures. However, an individual’s experiences and developmental environment may influence their perception of the musical interpretation of emotions (Morrison and Demorest 2009).
**Methods**

TransEmote generates music according to the emotional content in images. It does so in several steps in Fig. 1.

1. **Analyze input images and generate content descriptions.**
2. **Analyze the colors and use a color-to-emotion mapping to derive an emotional description.**
3. **Append the descriptions from (1) and (2) into an emotionally-descriptive profile.**
4. **Feed the distilled profile into a Natural Language Processing (NLP) section that maps the words to music-writing parameters.**
5. **Cycle the potential artefact through an intelligent quality check until it is satisfactory.**
6. **Output a playable audio file artefact.**

**Automated Content Extraction**

The content of images are strong indicators of an emotional profile. Whether it be the happiness felt at a wedding or the anger depicted between two people arguing, TransEmote is tasked with taking into account objects/actions taking place in images. In this stage, TransEmote extracts the content of images for later analyses. We set out to develop an application which combines both computer vision and natural language processing to create accurate and comprehensive captions from provided images.

The system achieved this through two methods: A convolutional Neural Network (CNN) for extracting features out of the image and a Recurrent Neural Network (RNN) for translating the extracted features into natural sentences. We trained our system on a massive captioned set of images from image-net.org. Our approach implements the Tensorflow library and Facebook’s PyTorch, which is an open-source machine learning library based on the Torch library. These are standard libraries used for applications such as computer vision and natural language processing. A CNN is used for feature extraction and can produce a rich representation of input images by embedding it into a fixed-length vector (Vinayals et al. 2014). This representation can be used for a variety of vision tasks, which makes it a natural choice to use a CNN as an “encoder” by using the last hidden layer as an input into an RNN. We utilized a VGG16 CNN architecture because of its overwhelming preferred use in the field of machine vision, which is outlined in Simonyan and Zisserman’s (Simonyan and Zisserman 2014) proposal of the system.

The images are then sent through the network and encoded into an array. The feature vector is linearly transformed to have the same dimension as the input to the RNN network. For our RNN model, which will “decode” the CNN output vector into readable text, we chose the standard long short-term memory (LSTM) network. This is a special type of RNN which is capable of learning long-term dependencies (Hochreiter and Schmidhuber 1997). For the RNN decoder training phase, the pre-trained CNN extracts the feature vector from a given image. After a linear transformation, the LSTM network input is the same as the CNN output. The decoder’s source and target texts are predefined. Using these source and target sequences with the feature vector, the LSTM decoder is trained as a language model conditioned on the feature vector.

**Color Analysis**

The next step of TransEmote’s emotional profiling abilities lie in the color analysis of each input image. From the developmental range of childhood, emotions are commonly associated with colors. In turn, we grow up to associate colors with emotions (e.g., *anger* is “red”, *happy* is “yellow”). Extracting prominent colors in an image will help in establishing what emotions the user is experiencing. Fugate and Franco’s (Fugate and Franco 2019) research into English speakers yielded a guide for major RGB values:

- **Anger:** Red (255, 0, 0)
- **Calmness:** Light Blue (0, 128, 255), Turquoise (0, 255, 255), White (255, 255, 255)
- **Disgust:** Sickly Green (204, 204, 0), Orange-Brown (204, 102, 0)
- **Fear:** Black (0, 0, 0), Red (255, 0, 0)
- **Happiness:** Yellow (255, 255, 0), Turquoise (0, 255, 255)
- **Sadness:** Navy Blue (0, 0, 255), Gray (160, 160, 160), Gray-Blue (0, 102, 204)

To extract the dominant colors, TransEmote uses a k-means clustering approach. We worked with three colors for this study so that we may focus results while still accounting for multiple emotional expressions.

We use a powerful machine learning library, scikit-learn, for our k-means clustering analysis. The RGB color scale forms a 3-dimensional vector space with orthogonal components. We can think of each pixel as lying somewhere in the 3D color vector space. After running this algorithm on our input image, we are able to derive a portrait of the 3 clustered, dominant colors in Fig. 2.

Once we have the RGB coordinates of our three dominant colors, we simply need to calculate the Euclidean distance to...
Figure 2: Example pixel colors extracted from the image in Fig. 4 shown in vector space.

Figure 3: Profiles of the three most dominant colors for the input image shown in Fig. 4.

The rest of the major colors, such as violet, were added without labels in order to avoid color values being matched to unreasonably distant emotional colors. This approach allows for the expression of multiple complex, and even disjoint, emotions that are commonly expressed through visual expressions (e.g. fear and calm). We simply have to append the emotions to the caption we previously generated.

Caption Distillation

Now the system needs to hand off this semantic information to the part of the system that maps it to musical parameters. First, we need to convert the captioned output into a friendlier format: an array of key words. To do this, TransEmote removes all duplicate and non-keyword strings from the generated caption. In doing so, we can better extract the key nouns and verbs that influence the image’s emotional tones in Fig. 4.

Music Generation and Critique

Semantic Evaluation

Semantic evaluation is critical to the way the system interprets and manifests the previous results. TransEmote uses a word distance approach (e.g. Word2Vec) as a generally intuitive, yet human-like way to transform input strings into meaningful musical context. Using a dictionary as our data structure, parameters can be added and [0, 1] scores retrieved, which can then be used to obtain real musical values such as pitch, range, and BPM. To improve on this approach, antonyms may be calculated (slow vs. fast) with a normalized score between them.

Instrumentation is also acquired using this approach. MIDI recognizes 128 unique instruments ranging from pianos, strings, brass, and synths. Each instrument is given a string representation and an average word-distance score is calculated for each input string. The instruments with the highest scores are used in the composition.

Rhythm

The length of a beat (quarter note) is established, then other rhythmic values are obtained through multiplication and division operations. TransEmote generates a rhythm by first providing it with a length, for example, $16 * q$ for a four measure rhythm. A sequence of note lengths are generated which add up to the length of the section, critiquing and validating each beat by constraining the options based on parameters discussed above. This establishes a baseline for rhythmic generation upon which tonal ideas are built.

Tone

Chord progressions provide a backbone to the melodic structure of music. TransEmote defines a progression in two parts: rhythm and intervals. A rhythm, as discussed above, is used to generate a pseudo-random sequence of intervals. Based on the obtained musical and global parameters, constraints are imposed on the random generation to improve the sense of intentionality, quality, and consistency. These include preferring 4/5/6 cadences, ending on the tonic, and not resting on unstable (e.g. diminished) intervals. With a chord progression in place, melody is laid over it in a similar constrained pseudorandom way. Rhythms are constructed to fit the length of each chord in the progression, with its notes defined by the chord itself.
Writing to MIDI

TransEmote uses the Mido Python library to handle the creation and editing of MIDI files. The aforementioned rhythms, chords, melodies, and instruments are converted into MIDI message format, and subsequently exported to MIDI, where it can be converted or used in a variety of ways.

Results

Method
We tested the success of TransEmote with a Google Form\(^1\) sent to universities and online AI communities. Our survey had eight examples in total with a five-point Likert scale asking “how well does the music convey the emotion of the image?” This was a blind study consisting of randomly organized test and control samples. The four test samples had the musical artefact created by TransEmote paired with their images. The control consisted of randomly generated artefacts paired with random images.

Analysis
Our survey received 109 full responses; which are ample data for effective analysis. We sum the times an artefact was given a four or five on each sample which count as successes. The results between the two groups are averaged. We found that 60.1% of test artefacts and only 20.4% of controls were rated by participants as ‘accurately portraying the emotions of the images’.

To corroborate this, a two population t-test suggests a highly significant (\(p < 0.0005\)) difference between the two groups. We found no significant differences between responses between genders.

Evaluation of Creativity

Beyond the ability to generate emotionally-relevant music, it is important that our system also engages in creative behavior. Prior work in computational creativity suggests that creative composition systems should be able to generate music that is novel, high quality, intentional, and typical of the domain (Boden 2009).

The creativity of TransEmote exhibits these attributes. As far as we know, the system is the first of its kind to extract semantic information from visual mediums and effectively translate it into musical audio. This reinforces the argument for novelty. Also, the emotional expression in music is a major portion of composition, and our system is able to bridge the non-verbal gap between these domains. Regarding quality, creativity doesn’t exist in a vacuum. We must involve the audience/community (Colton, Charnley, and Pease 2011). According to our survey results, the community at large does indeed ascribe value to the artefacts produced by TransEmote. The system also exhibits typicality because it utilizes existing domain rules of music composition and emotional expression. Lastly, our survey results prove the system is intentional. TransEmote sets out to extract visual semantics and uses them as an inspiration seed to generate author-guided, semantically-equivalent audio files. There is also an intelligent screening process to avoid artefacts that have already been generated.

\(^1\)shorturl.at/svMQU

Conclusion

We present a cross-domain system that extracts Western emotional profiles from images and translates them into music. Based on our survey, TransEmote has empirically demonstrated that it is able to usefully and creatively map and transform emotions between these domains.

For future steps towards improving the system, we would like to experiment extracting more information from images. Additionally, we believe that utilizing evolutionary algorithms would advance the emotional expression of musical generation. Training the thematic extractor on non-photographic images would also broaden its descriptive range. Lastly, we believe that TransEmote can be thought of as a preliminary support tool for music generation. Here, we implemented general models of musical expression but users have the ability to specify further musical parameters. Users would be able to integrate rules of jazz, for instance, in order to exclusively express semantic jazz music. We assert, in short, that TransEmote is computationally creative. The artefacts TransEmote produces are not mere generations, but thoughtful expressions of emotion. It is the authors’ hope that the next steps toward TransEmote-inspired systems will help to further enrich inspired music generation.

References


5. Performance
Show, Don’t (Just) Tell: 
Embodiment and Spatial Metaphor in Computational Story-Telling

Philipp Wicke
School of Computer Science
University College Dublin
Dublin, Ireland
philipp.wicke@ucdconnect.ie

Tony Veale
School of Computer Science
University College Dublin
Dublin, Ireland
tony.veale@ucd.ie

Abstract
To a human storyteller, a story is more than a textual artifact. Rather, as stories are both generated and generative, each is also a blueprint for performances to come. Tellers must draw on their own bodily affordances – from voice and gesture to movement around a stage – to bring stories to life, much as a conductor and an orchestra must translate a written score into actual music. This paper explores the creative challenge of translating from a story-text to a story-performance, from words to physical actions and characters to embodied actors. The mapping requires distinct models for gesture, narration, dialogue and stage direction if computer-generated tales are to transcend the limitations of their production process. Using the Scéalability framework, we evaluate the interlocking role of spatial metaphor and pantomime in turning a narrative artifact into a coherent performance.

All The World’s A Stage
Cinema is a visual medium, so filmmakers understandably live by the old Hollywood maxim, “Show, Don’t Tell.” Why use exposition or dialogue to tell of a dramatic event when you can show it directly on screen? Whether on stage or on screen, actors don’t just speak their lines; they live them out, with gesture, posture and meaningful spatial movement. The same can be said of storytellers of any kind. We humans put our backs into telling a story, so that our audiences can experience narrative events as though they were really there. Tellers are also performers, and their embodiment is key to the audience’s identification with the characters in a tale.

Mapping from a narrative text – of the kind typically produced by story generation systems – to an embodied performance requires a process not unlike the translation of a musical score into an orchestral performance. The composer’s vision for the piece must be respected, but difficult decisions that affect its realization must also be made. How many performers are needed, who will play what, and how must they be arranged, in space or in time? All the tricks of the theatre must be used to get the most out of what is available. When a piece has more roles than the cast has performers, the mapping is not an isomorphic one. As some roles are prioritized (or focalized) over overs, embodied characters must allude to the actions or feelings of those we cannot directly see or hear. Physical actors must show and tell, to convey a story world that is much bigger than them and their stage.

We focus here on the embodied realization of computer-generated stories with a mix of physical devices, specifically Amazon’s Echo/Alexa and the NAO anthropomorphic robot. Alexa serves as our omniscient narrator, while two NAOs, named Kim and Bap, speak and act as the story’s characters. Unlike past approaches to robotic storytelling, our robots are actors who take on specific roles in the narrative, and so they must act accordingly. Each robot speaks with a gender-appropriate voice for their character, making apt and often iconic gestures as they do, while moving about the stage in ways that metaphorically convey their relationship to other characters. In addition to speaking any dialogue given in the story itself, they must often add their own so that each physical action is paired with a naturalistic speech-act. They use this supplementary dialogue to comment on the tale itself, or to compensate for the physical absence of lesser characters.

In some ways, compensation is key to the whole process. Embodiment compensates for the weaknesses of the underlying story, by drawing our attention to how it is physically enacted rather than how it was automatically created. It may seem that clunky robots can do little to salvage a clunky tale, but each one adds to the innate comedy potential of the other. Consider Sunspring, an AI-generated sci-fi script that was filmed as a short movie by Ross Goodwin and Oscar Sharp. On the page, Sunspring is by turns absurd and seemingly un-filmable, with stage directions such as “He is standing in the stars and sitting on the floor.” Yet the human actors lean into
the script, using their faces, gestures and body-language to make the incongruous seem relatable on a human level. That audiences find the experience affecting and memorable has more to do with how it is embodied than how it was written. Our performance framework, named Scéalability, makes use of the Scéalextric story-generator (Veale 2017) for its tales, but it can, in principle, use any automated generator that provides access to a tale’s surface form and its deep structure. As in the case of Sunspring, our goal is to add coherence and entertainment value to computational stories by appealing to how we humans use gesture and space to create meaning.

This goal is unpacked in the following sections. We first explore research in automated story generation, and on how robots and other devices can be used to tell stories. We then present the Scéalability framework, with a specific emphasis on its models of dialogue, gesture and spatial metaphor. As our concerns go beyond system-engineering, Scéalability is also used to explore some hypotheses regarding the relative merits of iconic/pantomimic gesture and spatial metaphor in an embodied performance. Empirical studies are conducted to determine if audiences really can discern and appreciate the coherent use of space and gesture in the telling of a tale.

### Related Work and Ideas

#### Computational Storytelling

The generation of stories by mechanical means is a practice that predates AI and the advent of the modern computer. In 1928, the Canadian author William Wallace Cook marketed a system named Plotto, the master book of all plots, which gave budding authors more than 1600 plot schemas, cross-indexed for easy retrieval and recombination (Cook 1928). Cook’s system was just one of several that exploited nothing more sophisticated than the filing cabinet and the card index, placing the emphasis squarely on good data over algorithms. Yet, in retrospect, Plotto anticipates the AI approaches that would follow, from case-based-reasoning to story grammars.

Early AI approaches would be just as schematic as Plotto, while automating the tasks of planning and schema combination. One of the first, the Novel Writer system of (Klein et al. 1973), generated murder-mystery plots, while the more influential TALE-SPIN produced tales of woodland creatures by first building a world of goals and related characters for them to explore (Meehan 1977). Genre is also a tacit meta-schema in its own right, one that lends the weight of convention to otherwise insubstantial tales. So, just as the Universe system cranked out soap opera plots (Lebowitz 1985), the Minstreel system navigated a very different genre with just as many conventions: tales of courtly knights (Turner 1993).

The creative practices of human authors offer insights into how machines can write (or aspire to) tales of comparable quality. Dehn’s Author was the first AI system to explicitly model authorial goals in story creation (Dehn 1981), though more elaborate cognitive models have since been developed. The Mexico system of (Pérez and Sharples 2001) posits a two-phase cycle of creation called E-R, for Engagement-Reflection, in which the story generator alternates between bouts of incremental story development and subsequent consideration of the new opportunities that these may open up.

Cook’s development of Plotto in 1928 coincided with the flourishing of academic interest in the structuralist analysis of folk tales and similar cultural artifacts. In his Morphology of the Folk Tale, (Propp 1928) identified an inventory of recurring building blocks from which old tales are built, and from which new ones might be composed. His analysis remains relevant today, and forms the schematic basis for such generators as the PropperWryter system of (Gervás 2013).

These models are each symbolic in nature, and use logical forms that would pose little difficulty to the users of Cook’s Plotto book. In contrast, non-symbolic approaches sacrifice this interpretability for trainability, robustness and scale. For instance, the GPT-2 neural language model of (Radford et al. 2019) is trained on 40Gb of web text, and can “hallucinate” coherent continuations to arbitrary text story prompts. For instance, these authors show how continuations preserve the genre and guiding conceit of even speculative prompts, such as one that imagines the discovery of unicorns in the Andes. But GPT-2 and its kind are text-in, text-out generators that do not provide access to the plot-structure of the narrative text. While GPT-2 has hidden depths, its outputs are all surface.

**Relation to the previous research project** Scéalability needs access to the surface and deep structure of a story, so that it can choose gestures, dialogue and spatial movements to match the narrative intent behind the words. For this reason, we opt instead for the Scéalextric generator of (Veale 2017). Inspired by Cook’s Plotto, Scéalextric constructs its narratives from prefabricated segments of plot, which it connects end-to-end or expands top-down using recursive descent. Each plot segment comprises a sequence of transitive story verbs, from an inventory of 800 possibilities including fall_in_love_with, rescue and murder. Each verb has two participants, either the generic A and B or functionally-dependent roles such as A_spouse and B_friend. At heart, Scéalextric is a story-grammar for generating plots that are then rendered into a narrative text using an idiomatic mapping of story verbs to phrasal forms. It is its scale, ease of extensibility and modularity that distinguishes it most from other story generators. For instance, it provides a large database of famous characters for use in its stories, to instantiate the generic roles A and B (and their dependents) and to lend vivid colour (specific locales, weapons, vehicles, clothing, etc.) to the rendered text. Additional modules can also be inserted with relative ease, to attach physical gestures, spoken dialogue, and other stage directions so as to turn a narrative into a performance that shows as well as tells.

#### Embodied Storytelling

Story-telling with performing robots has been studied from a number of perspectives using robots of different kinds and varying physical affordances. The storytelling capabilities of an expressive robot face, called Reeti, was investigated by (Striepe and Lugrin 2017). In their comparative study, test subjects were presented with stories delivered by the Reeti, an audio book, and a neutral robot speaker. These authors use the same AttrakDiff questionnaire (Hassenzahl, Burmester, and Koller 2003) for their study as we shall employ in our own evaluation to follow. While the Reeti lacks a
body, it has an emotive face that is expressive enough to convey even an ironic intent (Ritschel et al. 2019). We shall use NAO robots that lack facial expression, with the expectation that their spatial mobility will compensate in other ways.

Storytelling is more than reading a text, as stories are enriched by the most basic visual cues. Early studies by (Heider and Simmel 1944) show that humans readily imbue simple geometric shapes that move about a screen with human-like intentions and mental states. Audiences are just as willing to attribute intention and emotion to anthropomorphic robots that purposefully gesticulate and stride about a stage.

Studies show that apt gestures can increase the expressiveness of an embodied storyteller (Csapo et al. 2012). Yet most robotic storytellers draw upon a small set of pre-defined gestures. For instance, the WikiTalk project of (Meena, Jokinen, and Wilcock 2012) relies on just seven gestures, which it uses to indicate discourse structure, as in the Open-Hand-Palm-Up gesture to mark the start of a new paragraph. WikiTalk uses a NAO robot to present the results of Wikipedia queries with a mix of voice and gesture, and shows how this integrated multimodality supports a natural interaction between human and machine (Jokinen and Wilcock 2014). But there have also been attempts to create custom gestures in order to suit an arbitrary speech context. Recent work by (Rodriguez et al. 2019) uses a Generative Adversarial Network (GAN) to produce apt gestures for a Pepper robot. Human gestures do not follow clear universal rules, so the generation of gestures is a non-trivial task. Nonetheless, there is some cognitive evidence for a schematic basis to many human gestures (Cienki 2005; Mittelberg 2018), and the generation of non-verbal behaviors for a virtual avatar based on such schemas is presented in (Ravenet, Clavel, and Pelachaud 2018).

Embodied Performance

Proxemics is the study of space for social interactions and their actors in a mutual environment. Research by Pope et al. investigated those interactions by theatre practitioners in physical spaces, using 360-degree filming and virtual reality (VR) (Pope et al. 2017).

Theatrical performances of robots on a stage require technological developments and a robust software framework. A focus on those low-level challenges, which require strong technical coordination has been presented in (Lin et al. 2009). In their study, a twin-wheeled, two-armed robot and a bipedal robot were set in a theatrical performance to show different performative challenges, e.g. story-telling concluding with a kiss between two robots.

A co-creative approach with a human and artificial performers has been investigated as improvisational theatre by (Mathewson and Mirowski 2017). The authors present two versions of AI-based chat-bots. Pyggy is an "Artificial Improvisor" using speech synthesis and speech recognition to communicate with an audience. It is capable of an open dialog interaction and Pyggy is embodied by a virtual avatar (a face with mouth movements synchronized to the speech). A different version of the robot called A.L.Ex. utilized Neural Language Model-based Text Generation to overcome Pyggy’s limited set of trained sentences.

In our research, we bring additional meaning from motion between our robotic actors, as well as orientation, pantomime and gesture in a study of space, taking other research, e.g. (Pope et al. 2017) into a computational domain. Our focus is not on these low-level challenges (Lin et al. 2009), but on simpler and more general uses of space and gesture in story-telling with improvisational elements between the robots.

Relation to the previous research project

Embodied storytelling within a generative framework has previously been studied in (Wicke and Veale 2018b; 2018a; Veale, Wicke, and Mildner 2019), who combined the Scéalextric story-generator with a single anthropomorphic NAO robot. Over 400 predefined gestures of the NAO are mapped onto almost 800 story verbs of the plot-generation system, so that a single robot teller makes an apt movement for every action it narrates. An interactive variant was later presented in (Wicke and Veale 2018a), in which the robot is guided through the story-space by a user’s answers to the robot’s questions. In a process that might be considered co-creative, the robot uses the Scéalextric plot graph to ask its questions and then branch according to the answer it receives. Once all questions have been answered, the plot is assembled and the tale is performed. A second device – the smart speaker Alexa/Echo – is added to the mix in (Veale, Wicke, and Mildner 2019). This pairing allows for banter between the devices, who now share the responsibility of narrating the tale (Alexa) and responding to it physically (NAO). It allows for a comparative analysis of the two devices, contrasting the disembodied voice of Alexa with the embodied antics of NAO. Although both contribute to the performance, it is always the embodied robot that adds the strongest humorous dimension. For this reason, Scéalability doubles down on its use of a robot by orchestrating the actions of two NAOs in showing and telling a tale.

The novel contribution of this research over the closely related research project is the addition of spatial movements by the robotic actors, which is baked into the storytelling system Scéalextric and coordinated by the Scéalability framework. Those changes and additions will be outlined in the next section. Moreover, an empirical evaluation presents its successful integration.

The Scéalability Embodiment Framework

Scéalability is conceived as an approach to storytelling-as-performance that places a definite emphasis on computer-generated narrative. This emphasis has a practical rationale: human-authored stories lack the semantic markup to allow performers to look beyond a surface text to see the plot logic within, while symbolic story generators offer transparent access to any level of the story at which the machine can reason. So, at the core of Scéalability sits a transparent generator of just this kind. Specifically, the structured outputs of the Scéalextric generator comprise a sequence of multilevel story beats, each of which contains the following elements:
1. A single narrative event, framed by a single story verb
2. Generic case roles for this story verb (e.g., A and B_{spouse})
3. The specific characters that fill these case roles (e.g., Neil Armstrong and Princess Leia, or C-3PO and HAL 9000)
4. A surface-level textual rendering for this single event, that incorporates character-specific details where possible
5. A logical connective that links this story beat to the next one (e.g., so, then, but, yet, because, and)

These strands are easily unpacked by Scéaleability so as to weave its own performative elements into a narrative. While any system that facilitates this separability of levels can be used as the generative heart of the framework, we evaluate the approach here with Scéalextric. Its various symbolic levels allow us to strengthen its existing narrative model while hooking in additional gesture, dialogue and spatial models. Let’s now look at each of those additional models in turn.

The Gestural Model
As shown in (Wicke and Veale 2018b; 2018a; Veale, Wicke, and Mildner 2019), storytelling can be gesturally-enhanced by mapping (almost) every story verb onto one or more gestures in the robot’s repertoire of physical flourishes. However, in those earlier systems, a single robot was expected to gesticulate for all of the characters in a story, with no regard for which character was making which gesture. With two robots, one for each of the two central parts A and B, gestures must be linked to specific case roles in each story verb so that they are performed by the right robot, and at the right time relative to the actions of other performers.

For a performance using n robots, we assume that only n characters will be embodied on stage. Human theatrical performances are more flexible than this, as real actors can play different parts in different scenes (with costume changes, accents and makeup to match). But our rigid NAO robots are not so flexible, and we wish to avoid confusing the audience with double-jobbing performers. So, with n = 2 robots, one can play role A as the other fills role B, and only the gestures associated with those roles are ever performed. The model must also indicate the order of gestures for a given verb (e.g., should A gesture before B for the story verb propose_to?), and whether they should be enacted before or after the narrator (in this case, Alexa) vocalizes its narrative text.

The Dialogue Model
Scéalextric does not have its own dialogue model, since its stories are rendered in a neutral third-person voice. While we can expect a robot’s gestures to speak for themselves, to an extent, such actions are rarely unambiguous, and it is just more natural for gestures to accompany live speech than omniscient narration. Moreover, spoken dialogue and physical action enrich each other when they are performed together.

The picture is complicated somewhat by the use of characters in supporting roles that are not embodied in the show. Roles such as A_{spouse} and B_{lawyer} have no robot presence, no gestures to perform, and no dialogue to utter, yet their presence in the narrative must still be felt by the audience. The model thus encompasses two kinds of dialogue: that which is uttered by embodied actors as they enact an event in which they appear, and that which they say to each other to comment on the unseen actions of other, disembodied roles. The former is embodied dialogue, the latter meta-dialogue.

Gestures and speech acts are expressions of the same urge to communicate, albeit in different modalities, so embodied dialogue is modeled in much the same way as physical gestures. For every story verb in the generator’s inventory, we simply define a set of apt vocalizations for the roles involved. Take, for example, propose_to: an actor in the agentive role may say “Will you marry me?” or even “It’s time to take our relationship to the next level,” while the actor in the patient role may reply “Wow, I don’t know what to say.” A reply must sound natural while being suitably vague, since the actors don’t yet know how the plot will unfold; the proposal may well be rejected in the next beat. In any case, all speech acts must be delivered in an appropriate sequence, and timed to enhance the gestures that are linked to the verb.

Meta-dialogue is a special case that comprises speech-acts that are uttered by actors in the central roles A and B, about characters that cannot be seen. Although the narrator tells us about these characters, the actors cannot show them to us. Suppose the next story beat is A_{Spouse} cheat with B_{Lawyer}. Our robot actors do not portray these characters, and cannot speak or gesticulate for them. Worse still, they have nothing to do or to say when the narrator speaks of these characters. Meta-dialogue allows A and B to talk to each other about A_{spouse} and B_{lawyer}, as though they were a Greek chorus. For example, A may say “I could kill that lawyer of yours,” to which B might reply “Just wait until you see the bill!” These jokes are baked into the dialogue model, as speech acts associated with the action A_{Spouse} cheat with B_{Lawyer}.

Our goal here is not to invent new speech acts, but to give our actors stock dialogue for events the generator can anticipate. Nonetheless, not every speech act is entirely scripted in advance. We allow our actors to ad-lib, by giving them underspecified dialogue of the form “You are +quality” or “You are −quality.” At the time it is spoken, +quality is replaced with a simile that accentuates quality, while +quality is replaced with one that ironically undermines it. For instance, +welcome may be replaced by “as welcome as a cool breeze on a summer’s day,” while +welcome can make way for “as welcome as a skunk at a garden party.” The system has a large stock of 1000s of similes to draw upon, but can also search on the web for fresh ones.

The Spatial Model
Our robot actors can do more than wave their arms and bend their legs; they can move about the stage in ways that meaningfully reflect their relationship to each other. Space is rich in metaphorical potential, so we speak of close ties and distant acquaintances, of losing touch and of coming together, of keeping our friends close and our enemies closer. These spatial metaphors are rooted in deep-seated image schemas (Johnson 1987) that conceptualize abstractions such as love and hate, trust and fear in experiential terms. A basic image-schematic model for reasoning about spatial metaphors was developed in (Veale and Keane 1992). Dubbed Conceptual Scaffolding, the model allows the semantics of non-spatial
verbs to be specified using spatial primitives such as up and down, connect and disconnect, and contain and release. By retrofitting this model onto Scéalextric’s story verbs, we can enable our robot actors to move about the stage in accordance with the actions they are enacting in the narrative.

We focus here on the connect and disconnect primitives, which allow us to signal the current state of the narrative via the relative closeness of the robot actors. Certain disconnect verbs, such as compete against, cause both to move apart, as each takes a step back, while asymmetrical verbs, such as resent and distrust, cause just one of the actors to move back. Similarly, some connect verbs, such as live with, cause both robots to move closer together, while others, such as spy on, cause just one to take a step closer, and yet others, such as pursue, cause one to move closer as the other moves away.

In every case, the unit of relative movement is a single step. Each robot begins the performance at a distance of six steps from the other. We hypothesize that audiences will register their spatial dance at a semantic level, for as the plot brings actors closer together and further apart, space will serve as a conceptual scaffolding for the twists and turns of the plot.

**Empirical Evaluation**

Scéalibility is designed to turn storytelling into a show. The narrative text of a story is augmented with spoken dialogue, gestures and stage directions for the actors to perform. Our major concern here is the value that embodied actors add to the telling, and so we focus on the relative contribution of gestures, which are often showy and pantomimic, and spatial movements, which more subtly achieve a cumulative effect.

We expect each form of embodiment to be more effective when used coherently – that is to say, in line with the plot – and to add to the audience’s appreciation of the story. We also expect their contributions to be additive, so that a performance with both is to be preferred over just one alone.

**A Pilot Study**

Each of our studies is based on the same Scéalextric story. Raters do not view the performances live, but watch video recordings of different settings. In a pilot study that we will only briefly summarize here, we showed our raters a recording of a complete Scéalextric story, which takes 3 minutes to view. All ratings are crowd-sourced on the Amazon Mechanical Turk (AMT) platform, and all questions are posed in a random order. We have paid 53, 52, 53 and 52 workers in 4 conditions (SpatialCoherent, SpatialIncoherent, PantomimicCoherent, PantomimicIncoherent). The length of this full story and its recording serves to dilute the effect of key actions and their embodied delivery. Nonetheless, the pilot shows that audiences prefer gestures a little more than spatial movements, and coherent over incoherent uses of embodiment. Statistically significant differences are found for each of these contrasts in user ratings on the AttrakDiff scale. Specifically, a post-hoc t-test shows a significant difference between Spatial Movement and Pantomimic Gesture ($p = 0.002$, Cohen’s $D = 0.094$) with means and standard deviations $\mu_{\text{Gesture}} = 4.591$, $\sigma_{\text{Gesture}} = 1.636$ and $\mu_{\text{Spatial}} = 4.430$, $\sigma_{\text{Spatial}} = 1.785$, and a significant difference ($p = 0.002$) in favour of coherent action (Cohen’s $D = 0.094$). Coherent spatial movement scores significantly better on the AttrakDiff scale than incoherent movement (Cohen’s $D = 0.272$).

**A Refined Study Protocol**

Since Scéalextric stories often contain many story beats, with various twists and turns, we build on the pilot study to show raters the following story excerpt that highlights just two story beats, which they rate on the same questionnaire:

- **A**=Hillary Clinton; **B**=Donald Trump;
- **B-friend**=Melania Trump; **N**=Narrator

**Gestures in a complete exhibition of a complete Scéalextric story, which takes 3 minutes to view. All ratings are crowd-sourced on the Amazon Mechanical Turk (AMT) platform, and all questions are posed in a random order.**

---

<table>
<thead>
<tr>
<th>Story Beat</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
</table>
| **Propose** | "I will release you." | "It's time we took our relationship to the next level. Will you marry me?"

---

**Additional information:**

- **N**: Thereafter Hillary would say that it was the other way around: that it was Hillary who dumped Donald!
- **B**: "Just let me go." A: "I will release you."
- **B**: "I will release you."
from the study. Furthermore, we only allow raters with a Master Worker Qualification on AMT to submit responses. This qualification is granted to those who have provided a large number of valid responses in previous tasks.

The same questionnaire is used in each study, and comprises 2 parts of 7 items each (excluding the 3 gold-standard questions). In the first part, raters answer 7 questions of the form ‘The performance of the robots is ...’ by choosing a value from 1 to 7 on these 7 AttrakDiff dimensions: (I) unpleasant/pleasant; (II) ugly/attractive; (III) disagreeable/likeable; (IV) rejecting/inviting; (V) bad/good; (VI) repelling/appealing; and (VII) discouraging/motivating. For the second part, raters signal their agreement with the following 7 statements with a value from 1 (strongly disagree) to 7 (strongly agree): (VIII) The robots appear human-like; (IX) The robots show their intentions; (X) I could act like one of the robots; (XI) The robot mirrored how I would react; (XII) I sided with one of the robots; (XIII) I am curious as to how the story continues; and (XIV) The robots’ movements are appropriate to events in the story.

**Study I: Coherence of Performative Elements**

Physical movements by robot actors can be eye-catching, and reinforce the embodied nature of the performance. But do they also add to the narrative in any semantic fashion? To test whether gestures and spatial movements are understood as meaningful contributions to the tale, we evaluate each under two conditions: the coherent condition, in which gestures or spatial movements are chosen to suit the semantics of each story verb; and the incoherent condition, in which gestures are chosen randomly, and spatial movements are performed contrary to the underlying image schema (so connect verbs are treated as disconnect verbs, and vice versa).

**Methods:** We present relevant parts of a sample story with both coherent and incoherent embodiment to human raters in a crowd-sourcing study on AMT. Two conditions (coherent / incoherent) for two embodiment strategies (gesture and spatial movement) necessitates four independent trials. Each rater is shown a 1-minute video that narrates the story with on-screen text and a synthesized voice-over. To focus their attention, the performance of just two story beats is presented on screen. These involve the story verbs propose to and release, which are rendered in the four trials as follows:

1. **Coherent Spatial Movement:** Robots move closer together (propose to) and later move further apart (release).
2. **Incoherent Spatial Movement:** Robots move further apart (propose to) and later closer together (release).
3. **Coherent Pantomimic Gesture:** Robot A bends its knee (propose to). Later robot B opens both its arms (release).
4. **Incoherent Pantomimic Gesture:** Robots A and B perform random pantomimic gestures (for each verb).

40 raters were recruited for each trial ($N = 40\times4 = 160$), and each was paid 0.40$ for completing the questionnaire.

**Analysis:** Human ratings for each trial were acquired over several weeks. Not counting excluded responses, the four trials elicited 29 valid responses for Coherent Gesture, 28 for Incoherent Gesture, 32 for Coherent Spatial Movement and 29 for Incoherent Spatial Movement ($N = 118$). For an overview of the statistical test results, see Table 1. The factor of coherence shows a significant p-value for an ANOVA test ($p = 0.000061$, mean squares $= 48.138$ and F-values $= 16.147$). A post-hoc t-test results in significant differences for all coherent and incoherent conditions with Cohen’s $D = 0.197$ (small to medium effect size) in favor of the coherent conditions. This is reflected in the average rating for all coherent conditions $\mu_{coherent} = 3.820$ and average rating of all incoherent conditions $\mu_{incoherent} = 3.480$. The coherent condition for both Pantomimic Gesture and Spatial Movement performs significantly better (Bonferroni-corrected $p_{\text{corrected}} = 0.001$ and $p_{\text{corrected}} = 0.047$, respectively) than the incoherent equivalent.

**Results:** Our hypothesis is thus supported, since we have shown that coherent uses of our two embodied narration strategies outperform the incoherent uses.

**Study II: Relative Value of Embodied Strategies**

In this experimental study, we focus on the value that coherent gestures and spatial movements add to a performance, whether individually (just one or the other) or both together.

**Methods:** This study evaluates three performance modes:

1. **Pantomimic Gesture:** the tale is performed with narration, dialogue and gesture, but no schematic spatial moves.
2. **Spatial Movement:** the tale is performed with narration, dialogue and schematic spatial moves, but no gestures.
3. **Combined Action:** the tale is performed with narration, dialogue, gesture and schematic spatial movements.

As before, in the Spatial Movement condition the robots face each other and move closer or further away as the plot progresses. The relative position of the robots at any time offers a spatial summary of their relationship status. For the Pantomimic Gesture condition, the robots do not alter their position in space, but do use iconic and showy gestures to communicate each story verb. For the Combined Movement condition, the robots apply both strategies together, i.e. they move to and fro as the plot demands, and they also make pantomimic gestures for each story verb in the plot.

The three one-minute videos are presented to 120 raters (or 40 for each) on the AMT crowd-sourcing platform. Each rater is shown just one of the three performances and then asked to evaluate it using our 14 + 3 item questionnaire. In return, each AMT rater is paid 0.40$ per questionnaire.

---

*See all of the recordings here: https://tinyurl.com/wpes3jl*
<table>
<thead>
<tr>
<th>Condition L</th>
<th>Condition R</th>
<th>L: Mean/Std</th>
<th>R: Mean/Std</th>
<th>p-value</th>
<th>Cohen D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space</td>
<td>Gesture</td>
<td>3.728/1.792</td>
<td>3.921/1.691</td>
<td>0.316*</td>
<td>-0.111</td>
</tr>
<tr>
<td>Space</td>
<td>Combined</td>
<td>3.728/1.792</td>
<td>4.131/1.762</td>
<td>0.002</td>
<td>-0.121</td>
</tr>
<tr>
<td>Gesture</td>
<td>Combined</td>
<td>3.921/1.691</td>
<td>4.131/1.762</td>
<td>0.206*</td>
<td>-0.111</td>
</tr>
</tbody>
</table>

Table 2: Post-hoc test for all three conditions. L and R denote conditions named in first (L) and second (R) column. *Bonferroni corrected p-value.

Analysis: All ratings were acquired over several weeks. Not counting excluded responses from those who failed the gold-standard questions, there are 32 valid responses for the Spatial Movement condition, 29 for the Pantomimic Gesture condition and 33 for the Combined Movement condition, for a total of \( N = 94 \) valid responses. An ANOVA reveals significant differences between the conditions, with \( p = 0.0019 \) (Sum of squares = 38.686, F-values = 6.292). A post-hoc t-test results in a significant difference between the Spatial Movement and Combined Movement conditions (\( p = 0.002 \) Bonferroni corrected). With a mean value \( \mu_{\text{Spatial}} = 3.728 \) and standard deviation \( \sigma_{\text{Spatial}} = 1.792 \) for Spatial Movement and a mean value \( \mu_{\text{Combined}} = 4.131 \) and standard deviation \( \sigma_{\text{Combined}} = 1.762 \) for Combined Movement, the effect favours the latter (Cohen’s D = 0.227). Pairwise comparisons of Spatial Movement/Pantomimic Gesture and Pantomimic Gesture/Combined Movement do not reveal any significant results. An overview of our analysis can be found in Table 2. Statistical tests have been conducted on the accumulated test construct (of all 14 items) and the results are visualized in Fig. 2. The whiskers indicate the standard error of the mean (\( \frac{\sigma}{\sqrt{N}} \)).

Results: Our findings suggest that a mix of embodiment strategies — what we have called the Combined Movement condition — is more appealing to viewers than Spatial Movement alone. However, there is no significant difference between the latter and Pantomimic Gesture. It seems that the subtlety of the actors’ image-schematic use of space is just as effective as their more showy pantomime actions, whether that is going down on one knee to propose, or making a servile bow to a dominant character. This result is important for more practical reasons too. Pantomime is achieved by a careful mapping of each story verb to one or more motor scripts in the robot’s repertoire. The results are eye-catching but often ad hoc, and depend more on cultural associations than semantics. In contrast, the robots’ spatial movements are governed by verb semantics, and follow generically from those semantics without the need for ad-hoc mappings.

But it must also be noted that Combined Movement does not significantly outperform pantomime on its own. Figure 2 shows that the margin of standard error around the mean for Pantomimic Gesture overlaps with that of the other two conditions. Even though the mean values \( \mu_{\text{Spatial}} = 3.728 \), \( \mu_{\text{Pantomime}} = 3.921 \) and \( \mu_{\text{Combined}} = 4.131 \) obey an ascending order, a significant statistical difference can only be found for the first and last of these. While a sample size of \( N = 94 \) is enough to show some effect, a future study on a larger scale should be more convincing on this front.

Conclusions and Future Work

The core insight of the presented research shows how spatial movement can be used to improve computational, embodied storytelling. So far, related works use a minimal set of scripted gestures tied to specific actions, which mostly show combinatorial novelty or mere generation. Scaling these storytelling performances to include new stories, new actors and more actions is problematic for scripted movements. Here, our approach of primitive motions shows its strength by allowing scaling with multiple actors, new stories and actions that only need to be associated on one dimension (connect / disconnect). We have provided empirical evidence that such approach is comparable with purely pantomimic performances (Study II).

With regards to interactive performance, we also see a role for gestures and spatial movements by the human audience. Robot performances are noisy affairs, and spoken dialogue must be timed so as not to overlap with the din of gears and servos in motion. So, what better way for the audience to convey their inputs to the story than by their own use of gesture and spatial metaphor? We are currently experimenting with visual analysis of the audience, and using emotion detection and pose estimation to recognize non-verbal inputs in the form of facial expressions (of surprise, anger, joy and sadness) and hand-gestures (thumbs up and down, rude finger gestures, aggressive fist motions, etc.). These won’t just allow audience members to naturally influence the story line. They will make the audience performers in their own right.

References

Cienki, A. 2005. Image schemas and gesture. From perception to meaning: Image schemas in cognitive linguistics


Creativity Metrics for a Lead-and-Follow Dynamic in an Improvisational Dance Agent

Meha Kumar, Duri Long, Brian Magerko

The Expressive Machinery Lab
Georgia Institute of Technology
Atlanta, GA 30332 USA
{mkumar77, duri, magerko}@gatech.edu

Abstract

Being creative in movement-based improvisational environments, such as dance floors, poses a difficult challenge for computers. It is particularly challenging for computers to judge the creativity of inputs and responses in these open-ended domains. LuminAI is an improvisational dance installation where an artificially intelligent agent dances with a user. In this paper, we enable the agent to judge the creativity of dance gestures through the development of creativity metrics, including the novelty, value, and surprise of a gesture. We then use these metrics to implement a lead-and-follow dynamic: when the user is performing less creatively, the agent tries to lead by performing more creatively. A user study is performed to compare the original and lead-and-follow systems, with results showing users found the lead-and-follow system’s gestures lower in quality than the original, more surprising than the original, and found the original system more engaging and influential on their actions.

Introduction

Though natural for most humans, being creative in movement-based improvisational environments, such as sports games and dance floors, poses a difficult challenge for computers. Agents with the ability to improvise movement could be used to inspire human choreographers, create more engaging training sequences in sports or physical therapy, or generate less predictable non-player character actions in video games.

One way to develop creative artificially intelligent (AI) agents is to equip them with an understanding of what is considered “creative”. Algorithms can measure the creativity of an artifact, such as a painting, poem, or dance move. The AI then uses this metric to guide itself as it explores the conceptual space, resulting in the generation of more creative artifacts.

Creativity can be defined in many ways, but one of the leading definitions used in the computational creativity community describes a composite score of three parts: the novelty, value, and unexpectedness of an artifact (Maher 2010; Boden 2004). Previous research has provided a basis for evaluating artifacts of creative systems (Maher 2010; Maher and Fisher 2012). However, in real-time interactive environments, efficiency is of utmost importance, so the metrics need to be adapted to produce creative results while not disrupting the experience.

One example of a research project where creativity metrics have been implemented in a real-time interaction is the Robot Improv Circus. The Robot Improv Circus is an installation in the domain of improvisational acting, where an agent plays an improv theater game with a user (Jacob 2019). Improvisational acting, however, requires general knowledge about the world to create an engaging performance. The difficult task of amassing knowledge for agents to use in human-agent interactions, or the knowledge-authoring bottleneck, becomes more evident as the number of possible meaningful actions increases. Since the space of improv theater gestures is essentially any action a human could perform in real life, the agent struggled to produce meaningful acts.

Comparatively, the domain of dance is simpler to interpret with computation, because it is more abstract (Jacob et al. 2013). Therefore, creativity metrics for an improvisational AI dance agent can be developed as a stepping stone towards a creative dance agent.

In this work, we will discuss the development of creativity metrics for an interactive dance installation called LuminAI (Figure 1). We present our implementation of these metrics, which uses the three-pronged definition of creativity described earlier to judge gestures.

Evaluating the creativity metrics can indicate whether they are useful in LuminAI and potentially other projects. Previous research with the Robot Improv Circus has shown that direct attempts to evaluate creativity metrics can be tricky (Jacob 2019). In that experiment, for each metric, participants were shown pairs of improv gestures and asked to select which one they thought would score higher for that metric. The results showed the developed metrics did not match human ratings. However, it also showed that the participants’ selections were close to random—meaning that experiments asking humans to directly rate gestures based on novelty, value, and surprise may be inherently flawed. Jacob (2019) suggests that having people compare two gestures without the greater context of an interaction may be too challenging. In an effort to determine whether the metrics are useful without directly asking about them, we have used the metrics to develop a potentially more engaging version of LuminAI.
Defining creativity

Boden (2004) defines creativity in three parts: novelty (how new the artifact is to the agent), unexpectedness (how surprising the artifact is in the current context), and value (the quality of the artifact). This definition has been widely explored within the computational creativity community. All three components are necessary when trying to design an AI agent which produces creative artifacts without human input. Agents which use only novelty require human input to guide them away from producing noise—in effect, the humans provide the metric of value (Kar, Konar, and Chakraborty 2015). Unexpectedness serves to account for artifacts which may have been seen before, but surprise us when presented in the current context.

Researchers have taken these three criteria and formalized them mathematically (Wiggins 2006), proposed possible algorithms to calculate these formalizations (Maher 2010; Lehmav and Stanley 2011; Maher and Fisher 2012), and implemented such algorithms in attempts to make creative agents (Jacob 2019). Another proposed element of creativity is typicality (how well the artifact conforms to the expectations of its domain) (Ritchie 2007); however, Jacob (2019) argues this metric is accounted for so long as the system does not maximize unexpectedness and novelty above all else. The metric of value is present to prevent this, and thus typicality would not provide any additional information. We use Boden’s definition in this work and develop a three-component algorithm for creativity based on novelty, value, and unexpectedness (referred to from now on as surprise).

Detailing each metric

Novelty. Novelty can be defined as how different an artifact is from other artifacts within the same domain that the observer has seen in the past (Boden 2004). Computationally, artifacts can be represented as vectors within some space, where the dimensionality of the space is determined by the number of features of the artifacts. Novelty is then intuitively understood as how far away an artifact is from all others, using any appropriate measure of distance.

One approach to determine novelty is to cluster the artifacts and determine to what degree the new artifact matches the nearest cluster (Maher and Fisher 2012; Barto, Mirolli, and Baldassarre 2013). Another approach utilizes Self-Organizing Maps in conjunction with clustering, which also reduces the dimensionality of the data and can provide useful visualizations (Maher 2010; Maher and Fisher 2012). A third approach involves determining the average distance from the artifact to its K-Nearest Neighbors, where K is some empirically defined natural number (Lehman and Stanley 2011; Maher and Fisher 2012).

In LuminAI, the gestures are represented as motion-capture data tracking each joint over time. This is an extremely high-dimensional representation, so any approach will require dimensionality reduction first. An effective pipeline for dimensionality reduction in LuminAI has already been produced (Liu et al. 2019). This pipeline re-
lies on feature reduction and Principal Components Analysis to compress the data to a few (typically 2 or 3) dimensions. Feature reduction refers to the condensing of a gesture’s original motion-capture representation into a smaller representation which does not include every single recorded frame. Specifically, the pipeline calculates the 15 most representative frames in any particular gesture, referred to as the keyframes of a gesture, and stores only those frames.

The reduction produced by this pipeline loosely groups gestures based on the major body parts involved in the movements (e.g., leg and hip movements are grouped together, left arm movements are grouped together). After viewing this reduced data in a visualization tool, it became clear that clustering is not an effective basis for measuring novelty because the data is sparse and the clusters are not separated enough for the valid novelty calculations. Therefore, we use average distance to the K-Nearest Neighbors on the reduced data as a measure of novelty.

Value Value can be defined as the usefulness, performance, and quality of an artifact to the observer, in the context of the surrounding culture. Clearly, value is highly dependent on the domain of artifacts being considered. We look at a definition of value that can be applied to any domain before focusing on value in dance gestures for LuminAI.

Maher and Fisher (2012) describe measuring value as similar to measuring novelty, that is, as a distance: this time in a “performance space.” In their application of measuring the creativity of laptop designs, they identified features relevant to the value of a laptop design by hand, creating vectors within a performance space. They then used a distance-to-centroid measure to determine overall value: laptops farther from the centroid of all the laptops were considered higher in value. Ideally, the agent will be able to determine the value of an artifact without human assistance. Therefore, we need to allow the agent to distill a dance gesture vector into its relevant features automatically.

The essential question here is: what features makes a dance gesture valuable? Montero (2012) suggests that the observer’s experience performing a particular dance style makes them a better judge of the quality of a movement. However, the agent in LuminAI has no perception of how it “feels” to perform a movement, so we measure value solely based on the agent’s visual perceptions.

Researchers have tried to understand why certain dancers are better than others through the lens of attractiveness (Neave et al. 2010; McCarty et al. 2017); they suggest that the motivation behind perceptions of “good” and “bad” dancing is reproductive. For example, McCarty et al. (2017) identify greater hip movement as one quality of a good female dancer which may indicate female mobility. One possible method of determining the quality of a gesture, therefore, is quantifying the amount of movement in certain key body parts. An aggregate score for quality may then be obtained from the various key body parts.

To avoid basing the value metric on attractiveness, which may introduce notions of gender and sexuality to the project, we also turn to a popular framework for understanding motion called Laban Movement Analysis (LMA) (Laban and Ullmann 1971). This framework was developed primarily for performers themselves, but also lends itself well to computational analysis. LMA interprets movement with four aspects:

- Body: what each part of the body is doing and how body parts are related and connected
- Effort: the qualities of the movement such as flow and weight
- Shape: the overall shape of the body and how it changes
- Space: the movement’s interaction with the surrounding environment

By using select LMA aspects as the relevant features for value, the agent could interpret movement using the same metrics that many human dancers use. A complication is that each style of dance may require its own calibration of the aspects. Since those interacting with the system in this study are novices who are not specialized in any style of dance, we will not use LMA to tune LuminAI’s value metric to any one style of dance. Instead, we will use the attraction-based definition of quality dance gestures, mitigating potential biases as much as possible. Future work could explore the use of LMA for specific dance styles.

Surprise Surprise can be defined as the unexpectedness of an artifact based on the observer’s expectations. As opposed to novelty, an artifact can still be considered surprising even if it has been seen before; surprise takes into account recent events which shape an expectation.

Barto, Mirolli, and Baldassarre (2013) define two ways to quantify surprise: 1) surprise as deviation from a prediction, and 2) surprise as the degree of difference between the agent’s beliefs before and after an event. The latter may be quantified using Kullback-Leibler divergence (Barto, Mirolli, and Baldassarre 2013). The difference between a predicted gesture and the observed gesture may be computed using a distance in the feature space, similar to novelty. Maher and Fisher (2012) consider an artifact surprising when it creates a new cluster within the conceptual space; this artifact changes the agent’s model of expectation significantly. In another work, Maher (2010) notes that surprise occurs when the observer has established a pattern, which the artifact then violates.

We utilize the definition of surprise as deviation from an expected dance gesture. We build expectations based on the last movement performed by the human and/or agent. In other words, we aim to find a dance move surprising if it deviates from the prediction built by the previous dance move.

Lead-and-follow human-agent interaction

Leading and following may create a more interesting user experience as the agent no longer solely responds to the user; it can also directly attempt to inspire the user. Lead-and-follow dance agents have been developed in the past with major limitations. Berman and James (2015) proposed a dance agent which dances with higher or lower intensity in response to its human partner. Our work is distinct in that...
the agent is less limited in its possible gestures, the agent uses creativity metrics as the basis for judging movement, and most importantly, the agent is able to both lead and follow the exchange. This may help the agent and human be true equals in the interaction, leading to a more stimulating experience for the user.

In the past, a lead-and-follow agent was built in LuminAI (Winston and Magerko 2017). This version of the LuminAI agent judged when to lead or follow based on the enthusiasm of a gesture (how wide or high-tempo it is) rather than the creativity of a gesture. The study found that users could tell the difference between the lead-and-follow and original versions, but that the original version was preferred. By using creativity metrics, we plan to build on this work by developing a lead-and-follow dynamic that is more engaging for users than the one implemented by Winston.

Implementation

In this section, we first describe the implementation of the creativity metrics (novelty, value, and surprise) themselves, and then how they are woven into a lead-and-follow strategy for the LuminAI agent.

Novelty

The gestures, represented as high-dimensional motion capture data, are reduced to two dimensions using a modification of the previously built dimensionality reduction pipeline described in Related Work (Liu et al. 2019). The modification focuses on the selection of the 15 keyframes. Rather than calculate the most representative keyframes of a gesture, which was inefficient, keyframes are chosen at uniform intervals from the gesture. This modification changed the chosen keyframes only slightly, leading to approximately the same reduction. The reduction seemed to preserve the pipeline’s ability to group gestures loosely based on which body parts were involved, while running much faster.

In this space, we calculate the novelty of a gesture as the average distance to its K-Nearest Neighbors (with K=5 yielding the best spread of novelty). The K-Nearest Neighbors algorithm utilizes an R-Tree of known gestures to quickly find neighbors. However, this novelty score would not be entirely useful for programming: average distances can only be compared to one another, and thresholds cannot be set to distinguish “high” and “low” scores. In order to scale the average distances to [0, 1], we pass them through an adaptive scaling tool. This tool dynamically sets the highest and lowest values it has seen thus far, allowing future numbers to be scaled to [0, 1] based on these values. This gives a final novelty score.

To avoid getting extreme novelty scores (i.e., 0 or 1) as the scaling tool sees its first few distances, a preprocessing step is needed. On startup, the system calculates the novelty of all gestures in the database, thereby passing all the unscaled scores through the scaling tool. If the database is reasonably varied, future gestures should not exceed the bounds set by the scaling tool too often.

Value

Following the definition of a quality dance move as based on attractiveness (Neave et al. 2010; McCarty et al. 2017), certain key indicators of good dance moves from both men and women are measured. By measuring the indicators for both men and women outlined by Neave et al. (2010) and McCarty et al. (2017), we hope to mitigate bias towards any particular gender when evaluating value. Specifically, we measure the amount of hip movement, shoulder movement, asymmetrical thigh movement, and asymmetrical arm movement from the motion capture data.

In order to measure these quantities, we determine the average amount of movement between consecutive frames. For efficiency, we first reduce the frame rate by half by removing every other frame. For hip movement, we calculate how far the left and right hip joints have moved between each pair of consecutive frames. These distances are summed to get a total amount of hip movement in the gesture. Then, we divide this by the number of frames in the gesture to achieve the average hip movement per frame. The average ensures that long gestures are not higher in value than shorter ones simply because they accumulate more movement. For shoulder movement, the same method is used, but we track the left shoulder, right shoulder, and neck joints instead.

For the asymmetrical thigh movement, we find the vectors representing the change in position for the right and left knee joints between each pair of consecutive frames. The left and right change vectors are subtracted to find a vector representing the asymmetrical movement between frames. We take the magnitude of this vector as our measure. As with hip movement, we sum this measure across all pairs of consecutive frames and divide by the total number of frames to achieve an average amount of asymmetrical thigh movement in the gesture. For asymmetrical arm movement, the same method is used, but we track the left and right elbows and wrists instead.

Once all four measures are found, we sum them together to get a total amount of valuable movement. As with novelty, this score needs to be scaled to [0, 1] to be useful. An adaptive scaling tool and preprocessing step are used in the
same manner. Gestures marked with high and low values are shown in Figure 2 and Figure 3, respectively.

**Surprise**

Following the definition of surprise as deviation from an expectation (Barto, Mirolli, and Baldassarre 2013), we must first define what the expected response gesture is when some dance move is performed. In order to truly know the expected response, a large data set would need to be collected consisting of many gestures with their observed response gestures. This would be a direction for future work, but for the initial development of the surprise metric in this paper, we make the assumption that the least surprising response would be mimicry. The next expected gesture, then, is the same as the current gesture.

Now, the deviation from the expected dance gesture can be defined. As Jacob (2013) describes, each gesture in LuminAI has certain key values associated with it, based on theories of movement, such as the energy, size, and tempo of the movement. We define surprise using the difference between the expected and actual gestures in two aspects: one to account for the difference in key values, and one to account for the difference in which body parts are used in the gestures. We use both because one can imagine a gesture that is surprising in only one aspect or the other. If a gesture consisting of large, fast arm circles followed a gesture of small, slow arm circles, it would be surprising (although the same body part was used). If a gesture of big, sudden kicks followed a gesture of big, sudden punches, it would be surprising (although the gestures have similar key values).

To obtain the part based on the key values, the difference between the expected and actual gestures’ key values are summed. This value is passed through an adaptive scaling tool. To obtain the part based on body parts, we reduce the dimensions of both the expected and actual gestures using the same dimensionality reduction pipeline used in the novelty calculation. In this reduced space, the distance between the two gestures is measured. This value is passed through another adaptive scaling tool. We chose to use separate scaling tools for the key value and body part components because the values they produce may be on wildly different scales, and we do not want one measure to dwarf the other when summed. The key values component and the body parts component are then summed and passed through a third adaptive scaling tool to obtain a final score for surprise.

**Lead-and-Follow Interaction**

Leading is added as an additional response mode (adding on to the existing modes of mimicking, random gesture recall, transformation, and related-gesture recall (Jacob and Magerko 2015)). The LuminAI agent selects from these response modes every time a user gesture is detected; it does not take into account whether it had been leading previously. Future work could explore incentivizing staying in leading mode for several turns.

First, the LuminAI dance agent must decide whether to lead based on the user’s gesture. If the measured novelty, value, or surprise of the gesture is low, it tries to lead. During development, the thresholds for leading were tuned until the agent led when the user performed known gestures, and followed when the user performed new or interesting gestures. Concretely, the agent leads when the novelty, value, or surprise scores of the user’s gesture are below 0.5. When leading, the agent selects a response gesture which has high surprise in the context of the user’s gesture. The agent finds the ten gestures with highest surprise from the database and randomly selects one to perform.

Ideally, the agent would perform the gesture with the highest value from the set of ten, but this response usually returned the same gesture repeatedly. This could be because gestures which measure high in value may be further away from other gestures in both the spaces used to calculate surprise (described earlier). Then, the same high-value gesture will be one of the ten highest-surprise gestures, and it will be returned every time the agent tries to lead. When not leading, the agent falls back on its existing response modes.

**Evaluation**

**Methodology**

In order to evaluate the system, a preliminary user study was performed. Subjects interacted with both the original LuminAI system and the creative lead-and-follow system described in this paper (referred to here as C-LuminAI) and provided feedback in the form of a survey and interview. Both systems were set to the discrete gesture mode described earlier, where only one party is dancing at a time. This ensures that the last gesture performed by either party is always known, which eliminates confusion about which gesture the dancers are responding to. Both systems’ agents were allowed to perform one gesture per turn. Both systems were pre-loaded with the same database of gestures. There were about 30 gestures, all recorded by researchers who are novice dancers. The systems were presented to subjects in a randomized order. Both interactions were video recorded.

First, subjects were allowed to familiarize themselves with the agent for a few turns, until the subject could successfully complete a gesture and see the agent’s response. Then, subjects interacted with each agent for about five minutes. Following the interactions, the users filled out a survey which asked about their perceptions of the agent’s dance moves, the levels of creative contribution and control both parties had over the interaction, their preferred agent, and the subject’s experience and comfort with dance. 5-point Likert scales were used to ask subjects about qualities of each system (e.g. “The dance moves the agent performed were good.” was asked using two Likert scales, one for each agent).

A short interview was then conducted to collect qualitative descriptions of the differences between the agents and reasoning behind the participant’s preferred agent.

**Results**

Seven subjects participated in the study, all of whom were college students with varying experience and comfort dancing. The small sample size means the results are not statis-
The dance moves the agent performed were good.

<table>
<thead>
<tr>
<th></th>
<th>Never</th>
<th>Sometimes</th>
<th>About half the time</th>
<th>Most of the time</th>
</tr>
</thead>
<tbody>
<tr>
<td>LuminAI</td>
<td>2</td>
<td>3.5</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>C-LuminAI</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

The dance moves the agent performed were surprising.

<table>
<thead>
<tr>
<th></th>
<th>Never</th>
<th>Sometimes</th>
<th>About half the time</th>
<th>Most of the time</th>
</tr>
</thead>
<tbody>
<tr>
<td>LuminAI</td>
<td>4</td>
<td>3.5</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>C-LuminAI</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

The agent kept me engaged in the interaction and made me want to continue dancing.

<table>
<thead>
<tr>
<th></th>
<th>Disagree</th>
<th>Somewhat disagree</th>
<th>Neither agree nor disagree</th>
<th>Somewhat agree</th>
<th>Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>LuminAI</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>C-LuminAI</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

I had an influence on the agent’s choice of dance moves.

<table>
<thead>
<tr>
<th></th>
<th>Never</th>
<th>Sometimes</th>
<th>About half the time</th>
<th>Most of the time</th>
</tr>
</thead>
<tbody>
<tr>
<td>LuminAI</td>
<td>3</td>
<td>3.5</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>C-LuminAI</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

The agent influenced my choice of dance moves.

<table>
<thead>
<tr>
<th></th>
<th>Never</th>
<th>Sometimes</th>
<th>About half the time</th>
<th>Most of the time</th>
</tr>
</thead>
<tbody>
<tr>
<td>LuminAI</td>
<td>3</td>
<td>3.5</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>C-LuminAI</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

How much did you contribute to the creative ideation of the dance interactions?

<table>
<thead>
<tr>
<th></th>
<th>No contribution</th>
<th>Less than the agent</th>
<th>Same as the agent</th>
<th>More than the agent</th>
<th>Sole contributor</th>
</tr>
</thead>
<tbody>
<tr>
<td>LuminAI</td>
<td>1</td>
<td>2</td>
<td>3.5</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>C-LuminAI</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

Discussion

The strongest result pointed to C-LuminAI giving nonhuman movements. This is likely because while leading, the agent was choosing among the most surprising gestures it could find, without taking the value of those gestures into account. This may have also led it to choose short gestures more often than LuminAI: the response modes of mimicking and transforming the user’s move are chosen more often in LuminAI and are guaranteed to be similar in length to the user’s gesture. That C-LuminAI gave strange gestures is actually promising when paired with the result that users found C-LuminAI more surprising. However, the agent seems to have veered too far off the surprising end of gestures into ones which limited the user’s ability to respond. Users felt that LuminAI was able to influence their behavior and engage them more with its better quality gestures. Interestingly, this did not correlate to LuminAI being preferred overall.

In Winston and Magerko’s (2017) study of their lead-and-follow agent in LuminAI, users preferred the original agent. However, the users’ reasoning for preferring one system to the other in that study were based on the increased mimicry of the original agent, while in our study, users’ reasoning was more based on the quality and variety of gestures performed. Both studies found some comments suggesting users enjoy the agent mimicking them. Based on these findings, a clearer lead-and-follow dynamic could allow users to gain the satisfaction of having the agent mimic them while the agent follows. Then, perhaps when the agent is leading, users may be more receptive to and less disappointed by the agent’s moves.

If the lead-and-follow dynamic is made clearer, and the gestures chosen when the agent is leading made more valu-
able, then the agent may reap the benefits of both versions. The lead-and-follow dynamic could be made clearer using text prompts, visual highlighting of which party is leading, and only using mimic and transform response modes when following.

The gestures chosen when leading were strange because C-LuminAI’s agent chooses one gesture randomly from the top ten most surprising gestures, but it ought to factor in value. This could be enabled by 1) vastly expanding the database of gestures so there are many high-value gestures that will often also be high-surprise when compared to the user’s gesture, or 2) changing the dimensionality reduction technique used on gestures so high-value gestures are not far away from all other gestures. This may not be possible, because better gestures may be inherently different from other ones. Either of these solutions would allow the leading mode to choose a high-value gesture from the top ten most surprising gestures, instead of randomly selecting one of the ten. The agent would then be able to be varied like C-LuminAI when leading, but would always have realistic gestures.

Overall, one of the most important factors in the user’s perception of this system is the quality of the gestures performed. The value metric described in this paper is a powerful tool for controlling the quality of gestures that are played back (or perhaps even stored into memory). One important addition would be factoring in the length of a gesture to the value metric. In addition, the surprise metric seems to successfully deliver more surprising gestures based on the survey results.

**Future Work**

Going forward, the developments described in this paper can be used to make LuminAI a more engaging system and to help explore the creative potential of the agent. The lead-and-follow dynamic developed in this paper can be improved and incorporated into LuminAI to reduce the repetition of gestures and provide an interesting new response mode. The creativity metrics can be used to explore creative arcs, as described by Jacob (2019). Creative arcs are paths the agent takes over its performance based on the creativity metrics: for example, the agent may start performing with high value and progress to low value, while also progressing from low novelty to high novelty. Overall, the ability of the agent to autonomously judge gestures for their creativity opens the door for new reasoning strategies and gesture selection algorithms.

**Conclusion**

In this work, algorithms were developed to measure the creativity of a dance gesture in the improvisational dance installation LuminAI. These metrics judged the novelty, surprise, and value of a gesture. These metrics can be used in the future to control the quality of gestures performed by the agent and add new reasoning strategies. The lead-and-follow dynamic developed in this paper improved the variety of gestures performed by the agent, but decreased their quality. Further development of this dynamic could increase user engagement with the system or help explore creative relationships between machine and human collaborators.

**Acknowledgements**

The authors would like to thank Lucas Liu, Cassandra Naomi, Swar Gujrania, and Jonathan Moon for their helpful discussions and continued work on the LuminAI system.

**References**


Jacob, M.; Coisne, G.; Gupta, A.; Sysoev, I.; Verma, G. G.; and Magerko, B. 2013. Viewpoints AI. In * Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*.


Towards Movement Generation with Audio Features

Benedikte Wallace¹, Charles P. Martin², Jim Torresen¹ and Kristian Nymoen¹

¹RITMO Center, University of Oslo
²Australian National University
benediwa@ifi.uio.no

Abstract

Sound and movement are closely coupled, particularly in dance. Certain audio features have been found to affect the way we move to music. Is this relationship between sound and movement something which can be modelled using machine learning? This work presents initial experiments wherein high-level audio features calculated from a set of music pieces are included in a movement generation model trained on motion capture recordings of improvised dance. Our results indicate that the model learns to generate realistic dance movements which vary depending on the audio features.

Introduction

Expressive movement is an intrinsic part of human life. Hand gestures, body language as well as dance can efficiently convey an emotional state. Simple movement patterns such as gait or arm movement can allow us to detect characteristics such as gender, personality or mood (Michalak et al., 2009; Pollick et al., 2001; Satchell et al., 2017). As such, a better understanding of body motion, and the analysis and generation of motion data is important to further develop fields such as human-robot interaction and human activity recognition. For dance movement in particular, generative models have potential as artistic tools for animation and choreography.

Research in embodied music cognition has identified several audio features that are relevant to how we move to music. Burger et al.’s (2013) work suggests that several mappings exist between different aspects of music and music-induced movement. The presence of a clear beat, for example, was shown to translate to faster movements of head and hands.

The work presented here is part of an ongoing research effort to examine how deep learning can be used to capture salient features of human movement, and especially dance movement, using full-body motion capture data and sound. As part of this work, we have collected a dataset of motion capture recordings of dance improvisation performed to six different musical stimuli. The improvisations are performed by experienced dancers and use contemporary music styles.

Here, we present the results of training a generative mixture density recurrent neural network (MDRNN) on our motion data and audio features which have been shown to affect certain aspects of movement to music. Without the inclusion of audio features, the MDRNN is able to generate sequences of movement which are (subjectively) realistic variations of the underlying training data. The results presented here indicate that the model retains this ability to produce movement variations when audio features are added. Our findings suggest that the model additionally learns that different audio features affect the way the body moves.

Motion Capture Data

Our dataset contains 54 one minute motion capture recordings of improvised dance performed by three experienced dancers. Each dancer performs three one minute improvisations to six different musical stimuli. The dataset was recorded using a Qualisys optical motion capture system with 12 Oqus 300/400 series cameras which capture 43 reflective markers worn by the dancers. Figure 1 shows how the 43 marker positions were reduced to a 22 point skeleton representation using the MoCap Toolbox 1.5 (Burger and Toiviainen, 2013). Small gaps in the data were spline-filled using Qualisys Track Manager 2019.3 and a 2nd degree Butterworth filter with a .03Hz cutoff was applied to remove any marker jitter.

Recordings in our dataset have been normalized so that the root marker (a weighted average of markers 41, 42, 6 and 7 in Figure 1) is centred at the origin. Body segment lengths are averaged across the three dancers ensuring that the data is invariant to global position and individual body dimen-
The data was captured at 240Hz and downsampled to 30Hz before model training to reduce the size of each example. The resulting 54 data tensors consist of 1800 frames (60 seconds at 30Hz) with 3-dimensional positions for each of the 22 points.

Two full motion capture recordings were withheld for testing while the remaining 52 examples were split into two sets, 80% were used for training and 10% for validation. Each example has been sliced into overlapping sequences of 300 frames and the spatial dimensions of each of the 22 points are scaled using min-max normalisation. The input to our model thereby consists of 78000 overlapping sequences of 300 frames with their corresponding audio features. The model performs a sequence-to-sequence mapping between the training examples (including audio features) and the shifted sequence of motion capture frames (excluding audio features – the model only predicts motion).

**Representing sound**

There are several aspects to consider when selecting appropriate audio features to represent the music examples to which the dancers were Improvising. Several previous works (Fukayama and Goto, 2015; Lee et al., 2019; Seo et al., 2013) have largely focused on the rhythmic, beat-matching aspects when generating dance.

Although the presence of a clear beat can affect our urge to move, moving in sync with a beat is only one of several ways musical features influence the way we move. For the experiments presented here, we have chosen to use two high-level rhythm- and timbre-related features: pulse clarity and sub-band spectral flux. Previous work by Burger et al. (2013) has shown connections between pulse clarity (Lartillot et al., 2008) and overall body movement, as well as sub-band spectral flux and movement of the head and hands.

Pulse clarity is a high-level feature which measures how clearly the underlying pulse of the music is perceived. Pulse clarity is estimated using the overall entropy of the energy distribution of the frequency spectrum within a musical piece. We calculate a series of pulse clarity values for each musical stimulus using a sliding window of 5 seconds and a hop size of 0.08 seconds. This gives us a time series wherein each value corresponds to a single frame of the motion capture data.

Spectral sub-band flux measures spectral changes in different frequency bands of an audio signal. Alluri and Toivainen (2010) found that the sub-band fluctuations in the region between 50 Hz and 200 Hz are related to the perceived “fullness” of a musical piece, while fluctuations in the region of 1600 Hz and 6400 Hz were linked to the perceived “activity” of the piece. These sub-bands also correspond to activity from rhythmic instruments such as kick drum and bass guitar for the lower frequency band and hi-hat and cymbals for the higher range.

We extract two frequency bands, one low-frequency band (50 Hz - 100 Hz) and one high-frequency band (3200 Hz - 6400 Hz) from the six musical stimuli. The spectral flux is then calculated using the same window and hop size as for the pulse clarity values resulting in a single sub-band flux value for each of the two bands for every frame of motion.

![Figure 2: Sampling from the MDRNN. One frame of mocap data and audio features is sent through the model. The MDRNN outputs the parameters of a mixture distribution which is sampled to generate the next frame.](image)

**Mixture Density Recurrent Neural Networks**

Mixture density networks (MDNs) (Bishop, 1994) treat the outputs of a neural network as the parameters of a Gaussian mixture model (GMM), which can be sampled to generate real-valued predictions. A GMM can be derived using the mean, weight and standard deviation of each component. The number of components needed to accurately represent the data is not known and is treated as a hyperparameter for our model. For the study outlined here, we have used 4 components. By combining a recurrent neural network (RNN) with an MDN to form an MDRNN we can make real-valued predictions based on a sequence of inputs. Figure 2 shows the model architecture of the MDRNN used in this work. The RNN consists of three layers of LSTM cells (Hochreiter and Schmidhuber, 1997). The three LSTM layers contain 1024, 512 and 256 hidden units respectively. The outputs of the third LSTM layer are in turn connected to an MDN. The LSTM layers learn to estimate the mean ($\mu$), standard deviation ($\sigma$) and weight ($\pi$) of the 4 Gaussian distributions of the MDN. This approach has the advantage of control over the diversity and “randomness” of sampling, and control over the number of mixture components that allow training to account for situations where multiple predictions could be considered equally suitable. MDRNNs have previously been applied to various other tasks such as sketches (Ha and Eck, 2017), handwriting (Graves, 2013), and music control generation (Martin and Torrensen, 2019).

To optimize an MDN, we minimize the negative log-likelihood of sampling true values from the predicted GMM for each example. A probability density function is used to obtain this likelihood value. This configuration corresponds to 8.5M parameters. The loss function in our system is calculated by the keras-mdn-layer Martin (2018) Python package which makes use of Tensorflow’s probability distribution package to construct the PDF. The model is trained using the Adam optimizer (Kingma and Ba, 2014) until the loss on the validation set fails to improve for 10 consecutive epochs.
The original movement sequence used to prime the model.

Movement generated using the priming sequence with corresponding audio features. Movements are less smooth and expressive than the original recording, but the generated movement follows the priming example nicely.

Sequence generated when the audio features from the priming example are replaced with features from a song not used in the dataset. The movements are more unstable and shaky.

Here, audio features are replaced with features calculated from a white noise signal. While the overall sequence is similar to the priming example, the movements contain more variation between frames, causing jittery movement.

Figure 3: These figures show the trajectories of hand and toe markers over time (left to right). When generating motion using audio which was not used in training (3c) or white noise (3d) the movements become more unstable.

**Altering Movement Using Audio Features**

When the MDRNN is trained on movement without the addition of audio features it is able to generate movements which are, under visual inspection, realistic. In this section, we examine the effect of altering the audio input for a model trained on both movement and audio data.

To examine to what extent the model has learned a correlation between the audio features and the movement we generate motion using a priming technique. When using priming the input to the model is taken from one of the examples which were withheld during training. At each time step, the input consists of the 3D positions of the 22 points and the corresponding set of audio features for that time step. The model then predicts the positions of the 22 points at the next time step. Thereby, the model always predicts the next pose using the values from the priming sequence. By altering the audio features of the priming example the output can be evaluated to determine the effect which different audio features have on the generated output. We examine three such cases here. First, we investigate a sequence generated using the audio features associated with the priming example itself, that is, features calculated from the musical stimuli the
dancer was improvising to when the priming example was recorded. Secondly, we replace this audio with an excerpt from a song which was not part of the training data. Finally, features calculated from a white noise signal is used to replace the original audio features. Figure 3 show keyframes from the 3 generated movement sequences as well as the motion sequence used to prime the model.

**Discussion**

When generating movement using the audio features belonging to the priming sequence (Figure 3b), the model generates movement which largely follows the example (Figure 3a). While the overall movement sequence is similar, some expressiveness seems to have been lost. The generated motion could be said to resemble a “dead pan” performance of the original sequence, as trajectories of arms and legs are to some extent muted in comparison to the original. This may be due to the model generalising movement across the training data.

Figure 3c shows the sequence generated when the original audio features are replaced with features calculated from a song not included in the training data. The model produces a movement sequence which matches the priming sequence well, indicating that the model is able to predict the next frame of the motion data even with unseen audio features. Still, additional noise is visible in this example (when compared to 3b), suggesting that the model has not fully learned to generate smooth movements when unseen audio features are used.

In the final figure, 3d, the audio features are replaced with features calculated from a white noise signal. Here, trajectories are decidedly affected by the audio. As with figure 3c and 3b the model still predicts reasonable positions for the 22 markers at every frame, but with a larger variation between frames, causing the resulting sequence to display jittery movement. This indicates that the model does rely on structured audio to generate realistic movements at the micro (if not macro) scale.

**Conclusions and Future Work**

These results indicate that the MDRNN model could be used to explore how music and audio features affect the way the dancers move, and how this manifests itself in the movements generated by a deep neural network. Much work remains to obtain a comprehensive understanding of how MDRNNs can model cross-modal interactions like those between sound and motion. An important aspect is how we can best evaluate the performance of this model. Finding good qualitative and quantitative ways to evaluate creative data generated by models such as this one will be a central question in our future work. Going forward, we will focus on systematically exploring metrics to evaluate the generated movements, train the model on a larger dataset and experiment with alternate audio representations.

**References**


Creativity Theatre for Demonstrable Computational Creativity

Simon Colton,1,2 Jon McCormack,2 Michael Cook1 and Sebastian Berns1

1 Game AI Group, EECS, Queen Mary University of London, UK
2 SensiLab, Faculty of IT, Monash University, Melbourne, Australia

Abstract
While the quality of computationally generated artefacts continues to improve, some people still find it difficult to accept software as being creative. To help address this issue, we introduce the notion of creativity theatre, whereby computational creativity systems demonstrate their creative behaviours, not only for the purpose of producing valuable artefacts, but also to heighten the sense that observers have of it being creative. We present an approach to this whereby an entirely separate AI system controls a casual creator app, which is normally used as a creativity support tool by people. We describe the ‘Can You See What I Can See?’ installation which performs such creativity theatre, and describe its operation in a recent open house event.

Introduction and Motivation
One of the accepted definitions of the field of Computational Creativity is given in (Colton and Wiggins 2012) as:

The philosophy, science and engineering of computational systems which, by taking on particular responsibilities, exhibit behaviours that unbiased observers would deem to be creative.

It is fair to say that most creative AI systems do not explicitly exhibit behaviours at all. That is, the processing they undertake in creating artefacts is often behind the scenes, and behaviours are either reverse-engineered (or guessed at) when people evaluate how the software has made its creations, or are read from technical papers describing how the software works. This lack of easily explainable and discernible behaviours makes it difficult for non-experts to independently assess the contribution particular systems/projects make towards advancing the field according to the definition above.

One way to partially improve the situation is for software to frame its work (Charnley, Pease, and Colton 2012) by outputting text which describes its motivations, how it makes artefacts, its evaluation of its products and processes, etc. A survey of framing in computational creativity research is given in (Cook et al. 2019). Another possibility is for creative software to more deliberately exhibit behaviours in a real-time fashion during its creative production. One way of achieving this is via anthropomorphisation, i.e., giving the software abilities which mimic human actions, even if they are not strictly required to create artefacts. For instance, in some installations, the Continuator music generation system (Pachet 2003) moves the keys of a real piano keyboard.

In general, an advantage of embedding a creative AI system in a robotic platform is such anthropomorphisation, as well as a clear separation of the creative behaviours of the system from the behaviours of the media being employed. For example, in robotic painting projects such as those in (Lindemeier et al. 2015) and (Tresset and Leymarie 2013), elements such as the robotic arm and camera mimic the hands and eyes of people, and as such, we can project particular behaviours onto them. These behaviours are separate from those of the analogue media (paints, brushes, pens, paper, etc.) that the robots employ. Similarly, in the You Can’t Know my Mind installation (Colton et al. 2015), The Painting Fool software makes paint strokes on-screen via a simulation of a hand holding a paintbrush. This anthropomorphisation enables viewers to separate The Painting Fool’s decision making system from its non-photorealistic rendering system. In artistic situations, such human-like physicality (whether real or simulated) can help audiences to project more nuanced behaviours onto creative AI systems, e.g., curiosity in (Gemeinboeck and Saunders 2010).

In projects like the ones described above, machines stage performances which, among other things, help express their creativity. To capture this notion, we introduce, motivate and explore below the notion of creativity theatre for such performances. We then describe a project where creative behaviours are explicitly foregrounded, with a separate AI system controlling a generative system. We conclude with a recap and details of future work, including an art installation based on the work presented here.

Creativity Theatre
The term security theatre was introduced in (Schneier 2003) to describe situations where security countermeasures are enacted to provide the feeling of improved security, while doing little or nothing to achieve it. The theatrics are intended to comfort members of the public and to serve as a warning to potentially malicious agents. With airport security as a particular focus, supporters of heightened security highlight the many times potential disasters have been averted, but critics argue that the measures aren’t effective, degrade passenger experience and have many unintended consequences, both in economic terms and with respect to more fatalities, for instance through increased car travel.

We can generalise the idea of pointedly and visually enacting a scenario, seemingly with a particular purpose, but also with an aim of changing public perception about some topic. Whether a particular type of enactment achieves the proposed purpose, and/or the changing of perception should be research questions subject to experimental evaluation in a
relevant domain. We are interested here in how AI systems can get to this experimental stage with respect to issues of computational creativity, i.e., how generative software can perform a form of creativity theatre in order to potentially heighten public perception of it being creative.

Being watched while being creative is, under most circumstances, not necessary for making a particular artefact, idea or process. Moreover, due to shared educational experience, people don’t necessarily need to show how they work in order for others to project aspects of creativity onto them. That is, we understand enough about the painting/design/composing/writing/ideation process to know that a person’s activities won’t be that much different to our own, even though they may be a virtuoso in their field (Pachet 2012). Moreover, there may be issues in such demonstrations demystifying the creative act, given that people tend to want to celebrate creative individuals as being special, and seeing them at work may normalise their behaviours.

Notwithstanding these points, the practice of creativity theatre in human endeavours is commonplace. Often, for instance, visual artists go out of their way to project aspects of their creativity on camera, e.g., Pablo Picasso painting on transparent glass (tinyurl.com/icccpicasso). Here, there is clear theatricality added to their standard artmaking techniques and in post-production of the films which promote their creativity. Other forms of art production have established theatrical outreach routes focused on creativity, with poetry slams, rapping competitions, improvisational theatre and musical improvisation being obvious examples. Watching people be creative is enjoyed as entertainment, as well as for enlightenment and inspiration. There are numerous films, online streaming channels, radio, television, online and in-print series following creative people. Streaming creative work has become particularly popular, with thousands of channels dedicated to live creative work on popular services such as Twitch.tv. Moreover, interviews with creative people are published, with their creative practices discussed and dramatised at length, e.g., (Peppiatt 2012).

In the artistic performances themselves, and in third-party reports thereof, there are often theatrical renditions of people being creative, with artificial elements of drama introduced and commonly juxtaposed with more mundane elements of production, which act as counterpoint. Further analysis reveals other commonalities, including the following:

- A sense of purpose in creating something new, albeit with progress often via exploratory and unexpected routes.
- Some unpredictability via improvisation and adaptation in the actions of the creative performer.
- Elements of virtuosity, presumably with the intention of adding to the feeling that the artist is special.

With such entertaining performances, the purpose is often to increase public perception of the performers’ creativity, rather than to produce a work of value. While it is not certain whether this actually works, this intention in clear.

In the context of computational creativity, implementations and installations are rarely conceived to promote the creativity of the system. Partly due to this, members of the public – who rarely read the technical papers describing AI systems – have generally formed their impressions in terms of what they see (or, in fact, don’t see), i.e., black box systems explicitly programmed by a person to undertake often simple tasks. In this context, it’s not surprising that arguments in favour of the creativity of an AI system often fall flat. Hence, the public are not being fully informed of advances in computational creativity, which may degrade their evaluation and adoption of the artefacts produced (Colton 2008). We therefore suggest that computational creativity systems are developed which can appropriately demonstrate their creative behaviours in a theatrical way, and we describe such a system in the next section.

**From Casual to Computational Creativity**

Casual creators are creativity support tools where user enjoyment is prized over productivity, fine-detailed control or the quality of output (Compton and Mateas 2015). They often have a generative element, and normally offer a straightforward user interface with instant and fun feedback that enables the searching of a space of novel artefacts rapidly and fluently. In many respects, their ease of use and rapid production of artefacts makes them ideal targets for third-party AI systems to control in order to explicitly exhibit creative behaviours for creativity theatre. Moreover, as casual creators are for human use, if people feel (somewhat) creative themselves while using the app, they may project notions of creativity onto a separate AI system controlling the app in similar ways, if the separation is properly communicated.

To explore the notion of creativity theatre empowered by casual creators, we implemented an AI system on top of the Art Done Quick casual creator app described in (Colton et al. 2020). This app employs a particle based image generation approach, where mathematical functions initialise, move and impose colours on particles, so that rendering shapes at particle positions in the appropriate colours produces an image. The functions, in addition to some rendering parameters, constitute a genome, with the rendered image being the phenotype. In overview, users make decorative imagery with the app via two main interfaces, as depicted in figure 1: (i) a sheet interface where a space of images can be explored through random generation and evolution, and (ii) an edit interface, where a single image can be altered.

Our long-term aim is to test whether creativity theatre encourages people to project notions of creativity onto an AI system controlling a casual creator app. The controller system we implemented for Art Done Quick was kept entirely separate from the base casual creator, i.e., the controller simulates taps, double taps and dragging on the iOS touchscreen as a user would, rather than accessing the relevant subroutines directly. Moreover, we made sure that the controller is not able to access any information that people cannot see. For public engagement purposes, with these measures, we are able to properly communicate the difference in Art Done Quick and the controller, and to emphasise similarities in how the controller and people use the casual creator app. To visually highlight these points, we added an animated gloved hand on top of Art Done Quick, as portrayed in figure 2. The hand taps, double taps and drags with one (simulated) finger,
Figure 1: Screenshots of Art Done Quick (iPhone version): (i) randomly generated images in the sheet interface (ii) one image selected for high-res rendering (iii) creating a montage using the edit interface (iv) adding a special effect.

and pinches with two fingers, in a similar manner and speed to the gestures that people employ when using the app.

Drawing on the above analysis of human creativity theatre, and noting that our goal is for people to watch the controller in action, we focused on an over-arching purpose for the AI control of Art Done Quick, namely to produce visual puzzles for people to try and solve. In particular, the controller uses the ResNet50 image classification system (Krizhevsky, Sutskever, and Hinton 2012) to project content predictions onto the images achievable through Art Done Quick. For some images, ResNet will have high confidence (of 80% or more) that it contains a particular object (such as a guitar) or a scene (such as a seascape). However, such images are rare, and finding them for presentation to users, accompanied with the question: “Can you see...”, was chosen as the overall purpose for the creativity theatre exercise.

Within this context, we identified and implemented in the controller AI system, simulations of the following subset of ways in which human users interact with Art Done Quick:

- Tapping an empty cell on the sheet interface, which fills it with a randomly generated image.
- Zooming and scrolling around the sheet to view image sets, then finding and inspecting ones of interest.
- Choosing and double tapping an image to produce 8 variations of it via genome mutation.
- Deleting images which are too similar to existing ones or undesirable in other ways.
- Editing a particular image through filters, transforms and collaging options available in the edit interface.

To implement these controlling actions, we employed a modified version of behaviour trees, which are normally employed to control non-player characters (NPCs) in videogames (Marcotte and Hamilton 2017). The first set of behaviours can be described as mundane, and includes tapping on empty cells to produce 25 randomly generated images. If the total number of images runs past 500, some are deleted to reduce the number to this threshold. To do this, images are k-means clustered using the analyses from a headless application of ResNet, then deletions are made evenly over the clusters. This may seem to break the maxim of not allowing the controller information (ResNet analyses)

that human users don’t have, but the machine vision functionality isn’t part of Art Done Quick, and the behaviour can be communicated as the controller using simulated eyes.

If at any stage, an image is produced which scores above 80% confidence by ResNet for an image category $C$, then the image is generated at high resolution and presented full-screen with the question: “Can you see...?” Some theatricality is introduced by giving audience members a 15 second pause in which they can try and guess what ResNet predicts the image to contain, after which the controller reveals the answer to the puzzle by changing the question to: “Can you see... a $C$?”, as in figure 2, presented for a further 15 seconds. The controller keeps a list of categories that it has previously used in the puzzles, and does not repeat the usage of one in a puzzle until at least an hour has past.

The second set of behaviours can be described as exploratory, and involve the controller hill climbing via mutation. That is, the controller occasionally does a sweep for any images scoring between 40% and 80% ResNet confidence. It chooses the highest scoring of such images and double taps it to produce 8 variations. If any of these improves on the previous score, it is mutated and so on, until either no improvement is seen, or a variation is produced which scores 80% or more, which – as before – is presented full-screen as a visual puzzle. The controller deletes any variation image that it doesn’t hill climb with.

The final set of behaviours can be described as directed, and involve the controller tweaking an image with ResNet...
confidence between 60% and 80% for a particular category. To do this, the controller invokes the edit functionality in Art Done Quick and systematically changes colour filters, texture overlays, lighting effects and liquifying transforms, with an example shown in figure 2. For each tweak, it records the ResNet confidence and returns to the highest scoring image for subsequent rounds of tweaking. If the tweaking ever produces an image of more than 80% confidence, the process stops and the visual puzzle is presented. Some unpredictability is introduced via random ordering of behaviours, but there is still too much predictable repetition. The above percentages were determined in advance, as sometimes – but not always – leading to visual puzzles.

For audience members to get a sense of what is going on, a banner is given at the top of the screen with single-word descriptions of the behaviour the controller is exhibiting, as portrayed in figure 2. We added theatricality by using provocative descriptions such as ‘gorging’, and making the controller take longer than necessary, to keep actions on a human scale. For instance, it has a ‘looking’ behaviour, where it simulates pinching on-screen to zoom out and view all the images on the sheet, followed by an artificial pause for 15 seconds to convey the idea that it is surveying its creations, as portrayed in figure 2. For a technical evaluation, the controller and Art Done Quick were run on an iPad Pro, with the screen mirrored onto a large screen during a period of six hours, as part of an open-house evening at SensiLab (sensilab.monash.edu). The performance ran without fail and produced more than 100 visual puzzles, which seemed regular enough to hold people’s attention.

Conclusions and Future Work

We have introduced the notion of creativity theatre as a loose analogy to security theatre and a tool for the demonstration of creative behaviours in a computational creativity setting. With an initial motivation and analysis of human creativity theatre, we identified some common aspects, including purposeful creation. We suggested adding secondary controlling AI systems to casual creator apps, as a way of achieving such theatricality. It is not clear yet if such secondary control will enhance the public perception of creativity in software, and we plan experiments to test this hypothesis. However, there is reason to be optimistic, given that casual creators level the playing field, i.e., AI controllers can produce similar artefacts in similar ways to people.

We are currently enhancing the controller to use more Art Done Quick functionality, including clustering, collaging and crossover of images (Colton 2012). Certain differences between behaviour tree usage for NPC control and usage for casual creation have become clear. We are currently developing a theory of creativity behaviour trees where unpredictability, purpose and virtuosity, among other aspects of human behaviour, are modelled. We expect this to enable us to implement more sophisticated elements of drama such as the controller seemingly changing its mind, expressing an emotional arc, and providing a commentary on what it is doing. These new behaviours will be developed with reference to various evaluation methodologies e.g., (Kantosalo and Riihiala).

The installation described above will be exhibited in 2021 at the VisionArtLas art exhibition in the Etopia Centre (zaragoza.es/ciudad/etopia), the theme of which is creative AI enhanced by machine vision. Titled ‘Can you see what I can see?’, we hope the installation will encourage people to realise that, while machine vision systems are generally developed to see as people do, they can also see differently and act as a second pair of eyes in an artistic setting. The installation will include a quiet area with seating where visitors will be able to play with Art Done Quick on its own. We hope this will encourage some visitors to question whether it is appropriate to project notions of creativity onto the controller, if they feel they are being creative, given that the software is doing very similar things to themselves.

Acknowledgements

We would like to thank the anonymous reviewers for much insightful feedback, some of which influenced this paper and all of which will be addressed in a longer account. We would also like to thank Marilía Bergamo for her design work for, and insights into, the SensiLab open-house installation.

References

Feel The Music: Automatically Generating A Dance For An Input Song

Purva Tendulkar\textsuperscript{1} Abhishek Das\textsuperscript{1,3} Aniruddha Kembhavi\textsuperscript{2} Devi Parikh\textsuperscript{1,3}

\textsuperscript{1}Georgia Institute of Technology, \textsuperscript{2}Allen Institute for Artificial Intelligence, \textsuperscript{3}Facebook AI Research

\textsuperscript{1}\{purva, parikh\}@gatech.edu, \textsuperscript{2}anik@allenai.org, \textsuperscript{3}abhshkdz@fb.com

Abstract

We present a general computational approach that enables a machine to generate a dance for any input music. We encode intuitive, flexible heuristics for what a ‘good’ dance is: the structure of the dance should align with the structure of the music. This flexibility allows the agent to discover creative dances. Human studies show that participants find our dances to be more creative and inspiring compared to meaningful baselines. We also evaluate how perception of creativity changes based on different presentations of the dance. Our code is available at github.com/purvaten/feel-the-music.

Introduction

Dance is ubiquitous human behavior, dating back to at least 20,000 years ago (Appenzeller 1998), and embodies human self-expression and creativity. At an ‘algorithmic’ level of abstraction, dance involves body movements organized into spatial patterns synchronized with temporal patterns in musical rhythms. Yet our understanding of how humans represent music and how these representations interact with body movements is limited (Brown, Martinez, and Parsons 2006), and computational approaches to it under-explored.

We focus on automatically generating creative dances for a variety of music. Systems that can automatically recommend and evaluate dances for a given input song can aid choreographers in creating compelling dance routines, inspire amateurs by suggesting creative moves, and propose modifications to improve dances humans come up with. Dancing can also be an entertainment feature in household robots, much like the delightful ability of today’s voice assistants to tell jokes or sing nursery rhymes to kids!

Automatically generating dance is challenging for several reasons. First, like other art forms, dance is subjective, which makes it hard to computationally model and evaluate. Second, generating dance routines involves synchronization between past, present and future movements whilst also synchronizing these movements with music. And finally, compelling dance recommendations should not just align movements to music, they should ensure these are enjoyable, creative, and appropriate to the music genre.

Figure 1: Given input music (top), we generate an aligned dance choreography as a sequence of discrete states (middle) which can map to a variety of visualizations (e.g., humanoid stick-figure pose variations, bottom). Video available at https://tinyurl.com/ybfakpxf.

As a step in this direction, we consider simple agents characterized by a single movement parameter that takes discrete ordinal values. Note that a variety of creative visualizations can be parameterized by a single value, including an agent moving along a 1D grid, a pulsating disc, deforming geometric shapes, or a humanoid in a variety of sequential poses.

In this work, we focus on designing interesting choreographies by combining the best of what humans are naturally good at – heuristics of ‘good’ dance that an audience might find appealing – and what machines are good at – optimizing well-defined objective functions. Our intuition is that in order for a dance to go well with the music, the overall spatio-temporal movement pattern should match the overall structure of music. That is, if the music is similar at two points in time, we would want the corresponding movements to be similar as well (Fig. 1). We translate this intuition to an objective our agents optimize. Note that this is a flexible objective; it does not put constraints on the specific movements allowed. So there are multiple ways to dance to a music such that movements at points in time are similar when the music is similar, leaving room for discovery of novel dances. We experiment with 25 music clips from 13 diverse genres. Our studies show that human subjects find our dances to be more creative compared to meaningful baselines.
Related work

Music representation. (Infantino et al. 2016; Augello et al. 2017) use beat timings and loudness as music features. We use Mel-Frequency Cepstral Coefficients (MFCCs) that capture fine-grained musical information. (Yalta et al. 2019) use the power spectrum (FFT) to represent music. MFCC features better match the exponential manner in which humans perceive pitch, while FFT has a linear resolution.

Expert supervision. Hidden Markov Models have been used to choose suitable movements for a humanoid robot to dance to a musical rhythm (Manfré et al. 2017). (Lee et al. 2019; Lee, Kim, and Lee 2018; Zhuang et al. 2020) trained stick figures to dance by mapping music to human dance poses using neural networks. (Pettee et al. 2019) trains models on human movement to generate novel choreography. Interactive and co-creation systems for choreography include (Carlson, Schiphorst, and Pasquier 2011; Jacob and Magerko 2015). In contrast to these works, our approach does not require any expert supervision or input.

Dance evaluation. (Tang, Jia, and Mao 2018) evaluate generated dance by asking users whether it matches the “ground truth” dance. This does not allow for creative variations in the dance. (Lee et al. 2019; Zhuang et al. 2020) evaluate their dances by asking subjects to compare a pair of dances based on beat rate, realism (independent of music), etc. Our evaluation focuses on whether human subjects find our generated dances to be creative and inspiring.

Dataset

For most of our experiments, we created a dataset by sampling ~10-second snippets from 22 songs for a total of 25 snippets. We also show qualitative results for longer snippets towards the end of the paper. To demonstrate the generality of our approach, we tried to ensure our dataset is as diverse as possible: our songs are sampled from 1) 13 different genres: Acapella, African, American Pop, Bollywood, Chinese, Indian-classical, Instrumental, Jazz, Latin, Non-lyrical, Offbeat, Rap, Rock ‘n Roll, and have significant variance in 2) number of beats: from complicated beats of Indian-classical dance of Bharatnatyam to highly rhythmic Latin Salsa 3) tempo: from slow, soothing Sitar music to more upbeat Western Pop music 4) complexity (in number and type of instruments): from African folk music to Chinese classical 5) and lyrics (with and without).

Approach

Our approach has four components – the music representation, the movement or dance representation (to be aligned with the music), an alignment score, and our greedy search procedure used to optimize this alignment score.

Music representation. We extract Mel-Frequency Cepstral Coefficients (MFCCs) for each song. MFCCs are amplitudes of the power spectrum of the audio signal in Mel-frequency domain. Our implementation uses the Librosa library (McFee et al. 2015). We use a sampling rate of 22050 and hop length of 512. Our music representation is a self-similarity matrix of the MFCC features, wherein \( \text{music}[i, j] = \exp(-||\text{mfcc}_i - \text{mfcc}_j||_2) \) measures how similar frames \( i \) and \( j \) are. This representation has been previously shown to effectively encode structure (Foot and Cooper 2003) that is useful for music retrieval applications.

Fig. 2 shows this music matrix for the song available at [https://tinyurl.com/yaurtk57](https://tinyurl.com/yaurtk57). A reference segment (shown in red, spanning 0.0s to 0.8s) repeats several times later in the song (shown in green). Our music representation captures this repeating structure well.

Dance representation. Our agent is parameterized with an ordinal movement parameter \( k \) that takes one of 1,..., \( K \) discrete ‘states’ at each step in a sequence of \( N \) ‘actions’. The agent always begins in the middle \( \sim \frac{k}{2} \). At each step, the agent can take one of three actions: stay at the current state \( k \), or move to adjacent states \( (k - 1) \) or \( (k + 1) \) without going out of bounds. We explore three ways to represent a dance.

1. State-based (ST). Similar to music, we define our dance matrix \( \text{dance}_{\text{state}}[i, j] \) as similarity in the agent’s state at time \( i \) and \( j \): distance between the two states normalized by \( (K - 1) \), subtracted from 1. Similarity is 0 when the two states are the farthest possible, and 1 when they are the same.

2. Action-based (AC). \( \text{dance}_{\text{action}}[i, j] \) is 1 when the agent takes the same action at times \( i \) and \( j \), and 0 otherwise.

3. State + action-based (SA). As a combination, \( \text{dance}_{\text{state+action}} \) is the average of \( \text{dance}_{\text{state}} \) and \( \text{dance}_{\text{action}} \).

Reasoning about tuples of states and actions (as opposed to singletons at \( i \) and \( j \)) is future work.

Objective function: aligning music and dance. We use Pearson correlation between vectorized music and dance matrices as the objective function our agent optimizes to search for ‘good’ dances. Pearson correlation measures the strength of linear association between the two representations, and is high when the two matrices are aligned (leading to well-synchronized dance) and low if unsynchronized.

For an \( M \times M \) music matrix and \( N \times N \) dance matrix (where \( N = \text{no. of actions} \)), we upsample the dance matrix to \( M \times M \) via nearest neighbor interpolation and then compute Pearson correlation. That is, each step in the dance corresponds to a temporal window in the input music.

In light of this objective, we can now intuitively understand our dance representations.

State-based (ST): Since this is based on distance between
states, the agent is encouraged to position itself so that it revisits similar states when similar music sequences repeat. Note that this could be restrictive in the actions the agent can take or hard to optimize as it requires planning actions in advance to land near where it was when the music repeats.

Action-based (AC): Since this is based on matching actions, the agent is encouraged to take actions such that it takes the same actions when similar music sequences repeat. This has a natural analogy to human dancers who often repeat moves when the music repeats. Intuitively, this is less restrictive than ST because unlike states, actions are independent and not bound by transition constraints; recall that the agent can only move to adjacent states from a state (or stay).

**Search procedure.** We use Beam Search with a single beam to find the best dance sequence given the music and dance matrices, as scored by the Pearson correlation objective described earlier. We use chunks of 5 dance steps as one node in the beam. The node can take one of $3^5$ values (3 action choices at each step). Specifically, we start with the first 5 steps and the corresponding music matrix (red boxes in Fig. 3). We compute Pearson correlation with all $3^5$ dance matrices, and return the best sequence for these 5 steps. Next, we set the first 5 steps to the best sequence, and search over all combinations of the next 5, i.e., $3^5$ sequences, each of length 10 now. See orange boxes in Fig. 3. This continues till a sequence of length $N$ has been found (i.e., the music ends). Our music and dance representations scale well with song length. Our search procedure scales linearly with number of actions, and we discuss approaches to overcome this in Future Work.

**Baselines.** We hypothesize that dances that have a balance of surprise and value will be perceived to be more creative. That is, dances where an agent moves predictably or that are not synchronized with the music will be deemed less creative. This motivates our baselines:

1. Synced, sequential (SS). The agent moves sequentially from one extreme of the state space to the other till the music ends. It only moves when there is a beat in the music. The beat information is extracted using the implementation of (Krebs, Böck, and Widmer 2015) from the Madmom library. This baseline is synced, yet predictable and uninteresting.
2. Un-synced, sequential (US). The agent also moves sequentially, but ignores the beats and moves at every step. This baseline is unsynced and predictable.
3. Synced, random (SR). The agent takes a random action from its allowed actions at every beat, and stays put otherwise. This baseline is synced and quite unpredictable, so we expect this to be more interesting than SS and US.
4. Un-synced, random (UR). The agent takes a random action from its allowed actions independent of when the beat occurs. This baseline is unsynced and unpredictable.

**Evaluation via human studies**

We compare our approach using the 3 dance representations against the 4 baselines for 25 song snippets and values of $N \in \{25, 50, 100\}$ (no. of steps in the dance). We set the number of states an agent can be in to $K = 20$. For this experiment, we visualize the agent as a dot, with state indicating a location on a 1-D grid. We first compare approaches with the same $N$, and then for the best approach, compare different $Ns$. We then compare our best approach to the strongest baseline using other visualizations to evaluate the role visualization plays in perception of dance creativity. For each of these settings, we show subjects on Amazon Mechanical Turk (AMT) a pair of dances and ask them: Which dance (1) goes better with the music? (2) is more surprising/ unpredictable? (3) is more creative? (4) is more inspiring? Subjects can pick one of the two dances or rate them equally.

**Dance representation.** The 7 approaches amount to $\binom{7}{2} = 21$ pairs of dances per song per $N$. We showed each pair (for the same song and $N$) to 5 subjects on AMT. For the 25 songs and $N \in \{25, 50, 100\}$, this results in a total of 7875 comparisons. 210 unique subjects participated in this study.

See Fig. 4. Table cells show win rate of approach in row against approach in column. Subscripts and green shades show statistical confidence levels (shown only for $> 80\%$). For example, dances from SR are found to be more creative than those from SS $61\%$ of the times. That is, at our sample size, SR is more creative than SS with $99\%$ confidence. Among baselines (rows 1 to 4), humans found random variants (SR, UR) to be more unpredictable (as expected), and UR to be more creative, better synchronized to music, and more inspiring than sequential variants (SS, US). UR is the best-performing baseline across metrics. We hypothesize that UR performs better than SR because the latter only moves with beats. Comments from subjects indicate that they prefer agents that also move with other features of the music. All our proposed approaches perform better than SS, US, SR baselines across metrics. AC is rated comparable to ST and SA in terms of (un)predictability. But more creative, synchronized with music, and inspiring. This may be because as discussed earlier, state-based synchronization is harder to achieve. Moreover, repetition in actions for repeating music is perhaps more common among dancers than repetition in states (location). Finally, our best approach AC is rated as more creative than the strongest baseline UR.
Figure 4: Evaluation via human studies of dances on 4 metrics – a) creativity, b) synchronization with music, c) unpredictability, and d) inspiration. Table cells show win rate of approach in row against approach in column. Shade of green and subscript shows statistical confidence (only for > 80%).

**Number of steps.** With a higher number of steps, the agent can sync to the music with higher precision. However more steps would add more “jumpiness” to the dance, which may not be desirable. We evaluate our best approach (AC) for $N \in \{25, 50, 100\}$. This gives us $\binom{3}{2} = 6$ pairs of dances per song. We showed each pair for each of the 25 songs to 5 AMT subjects; 375 pairwise comparisons from 22 unique subjects. Subjects find dances with 100 steps to be more creative than 50 and 25 steps at 99% statistical confidence, with 100 steps preferred 69% and 73% of the times respectively.

**Effect of visualizations.** Finally, we analyze how choice of visualization affects perception of dance creativity. We compare our best approach (AC) with the strongest baseline (UR) for 6 different visualizations including a pulsating disc, a stick figure, and collections of deforming geometric shapes. Including the dot on a 1-D grid from earlier, we have 7 pairs of dances for 25 songs and 5 evaluators; 875 comparisons from 59 unique subjects. Preference for our approach in row against approach in column. Shade of green and subscript shows statistical confidence (only for > 80%).

**Discussion**

Our preliminary study with a simple agent gives promising indications that subjects find dances discovered using our flexible, intuitive heuristic to be creative. The next step is to train more complex agents to dance. Our search-based approach will not scale well with larger action spaces. We plan to use machine learning approaches to optimize for the music-dance alignment, so that given a new song at test time, an aligned dance sequence an be produced without an explicit search. Rather than supervised learning approaches described in Related Work which require annotated data, we will explore Reinforcement Learning (RL) using our objective function as a reward. This retains the the possibility of discovering novel dances, which is central to creativity.

**References**

Appenzeller, T. 1998. Evolution or revolution?


Jacob, M., and Magerko, B. 2015. Interaction-based authoring for scalable co-creative agents. In ICCC.


Manfré, A.; Infantino, I.; Augello, A.; Pilato, G.; and Vella, F. 2017. Learning by demonstration for a dancing robot within a computational creativity framework. In IRC.


6. Philosophy and Evaluation
A Leap of Creativity: From Systems that Generalize to Systems that Filter

Porter Glines, Brandon Biggs, and Paul M. Bodily
Department of Computer Science
Idaho State University
921 S. 8th Ave, Pocatello, ID 83209 USA
glinport@isu.edu, biggbran@isu.edu, bodipaul@isu.edu

Abstract

In his work “Mere Generation: Essential Barometer or Dated Concept?”, Ventura (Ventura 2016) categorizes creative processes along a spectrum of increasing creativity. While the spectrum provides insight into the dimensions through which creativity can be augmented, it does not of itself provide insights into how to advance a system through these dimensions. In this paper, we present some theoretical and practical insights on advancing along one commonly problematic rung of this ladder, namely from a system that exhibits generalization (i.e., the ability to generalize beyond an inspiring set) to a system that exhibits filtration (i.e., the ability to self-evaluate and filter results). One potential challenge in this transition is that filtration requires having a sufficiently large number of solutions to filter from the generalizing model. We propose that one solution to this problem is achieved not through increasing the size of the inspiring set (an obvious solution that brings additional problems), but rather through amplifying the generalization of the system to produce a greater set of novel artefacts to filter. We compare a new version of a system, NhMMonic, for generating creative mnemonic devices with a new conceptualization model that allows greater generalization. We demonstrate how filtration, which was not possible in the early version of NhM-Monic, only becomes feasible with the more generalizable model.

Introduction

The field of Computational Creativity (CC) has been supported in its quest by several significant contributions in the domain of CC theory. One such contribution exists in Ventura’s spectrum of creative systems (Ventura 2016). This spectrum suggests that there exist at least seven different levels along the path towards computational creativity including levels such as randomness, memorization, generalization, and filtration (see Figure 1). Ventura asserts that along this spectrum, real computational creativity starts at least as early as generalization with filtration representing perhaps a conservative threshold.

While this spectrum is useful for measuring the progress of applied CC systems, it leaves two important questions unanswered:

1. For each level of the spectrum, what challenges are CC systems likely to encounter?

2. What suggestions can be made to overcome those challenges?

Answers to these questions would provide a way to actualize the spectrum into a guide for augmenting the creativity of computational systems.

Our motivation in considering these issues came about in the context of our previous work using constrained Markov models to generate mnemonic devices (Bodily, Glines, and Biggs 2019). Markov models are an example of a generalizing model. The application of constraints to Markov models represents the act of filtration. In applying constraints to generate mnemonic devices, it frequently occurred that no satisfying solutions could be found.

The purpose of this paper is to provide answers to two questions stated above with specific regard to systems that have achieved the level of generalization and are attempting to make the “leap” to the level of filtration. This step is of interest as it marks the transition from a budding creative system to an intentionally creative system. This leap is significant in light of the fact that of the last four levels of the spectrum—where true creativity is said to emerge—this is the first step.

Generalization systems produce artefacts using an internal conceptualization—a model which embodies an understanding of a domain and allows for the creation of artefacts that belong to the domain (Ventura 2017). Examples of conceptualizations include using long short-term memory models for music generation (Nayebi and Vitelli 2015), neural networks for visual art (Norton, Heath, and Ventura 2013), and Markov processes for music and text generation (Pachet, Roy, and Barbieri 2011; Barbieri et al. 2012).

One particular challenge we have repeatedly observed in the development of CC systems at this level is the challenge of dealing with diminishing solution spaces. This problem arises commonly when attempts are first made to add filtration to a generalization system because filtration by definition implies the reduction of a system’s solution space. The purpose of the filtration step is to equip the system with self-evaluative capabilities for restricting the artefacts it generates based on measurements of fitness. However, a well-known trade-off arises: stricter filtering leads to better, but
Figure 1: Ventura’s (2016) spectrum of creative systems provides a means by which to measure the progress of a system towards becoming creative. Characterizing challenges and solutions that are specific to each level in the spectrum helps to actualize the spectrum into becoming a guide for building more creative systems.

fewer results. In some cases the results are so few that it becomes difficult to justify that the system is capable of generating anything, let alone artefacts that are novel. How can systems overcome this challenge?

A simple solution for increasing the solution space is to simply increase the size of the inspiring set. For many conceptualizations of CC systems this alone will increase the overall throughput of the system, and often increases the generalizability of the system as well. However, for most domains, finding a larger inspiring set ranges from being impractical to an impossibility. What more practical solutions exist?

We propose and illustrate through example how increasing the generalizability of a generalization system through abstraction and regularization can increase the solution space without requiring a larger inspiring set. Well-known methods exist for generalization of most conceptualization models used for CC systems, including L1 and L2 regularization for neural networks, shortening the Markov window length in Markov processes, generalizing the fitness function for genetic algorithms, and abstracting rules in rule-based systems. Through regularization and abstraction, a system is able to better leverage the knowledge in an inspiring set in order to increase the solution space.

In demonstrating the impacts of abstraction and generalization, we comparatively consider the performance of two models: a less abstract model (CoMP) and a more abstract model (CHiMP). We assess the ability of each model to intentionally produce novel artefacts. We choose to focus explicitly on the creative attribute of novelty—setting aside the attributes of value and intentionality—inasmuch as it is the attribute of creativity most directly relevant to our discussion (Ritchie 2007; ?). We discuss the impacts of generalization on value in the discussion section below.

Methods

NhMMonic (Bodily, Glines, and Biggs 2019), is a CC system designed to generate mnemonic devices. At its heart, NhMMonic uses a constrained Markov process (CoMP) for its conceptualization model. This constrained Markov process allows for the combination of a (non-hidden) Markov process (e.g., trained on words) and a set of unary constraints (e.g., word-starts-with constraints) such that the model is able to generate constraint-satisfying sequences according to Markovian probabilities (Pachet, Roy, and Barbieri 2011). In previous work we demonstrated through qualitative surveys the strength of this model (particularly at higher Markov orders) for generating effective mnemonic devices. A byproduct of our analysis revealed that for many mnemonic device problems, the addition of constraints (i.e., filtering) resulted in NhMMonic being incapable of finding satisfying solutions despite being trained from relatively large inspiring sets.

A known method for increasing the generalization of Markov models is through the introducing of an abstract hidden layer resulting in a model known as a hidden Markov process. Direct dependencies between adjacent observed sequence elements are dissolved in the hidden Markov process, allowing for greater decoupling between sequence elements. This generally results in hidden Markov processes having significantly higher expressivity with respect to their non-hidden counterparts.

To combat the challenges facing NhMMonic with respect to a diminishing solution space, we designed a new conceptualization model for the system that combines hidden Markov processes with constraints in much the same way that constrained Markov processes combined non-hidden Markov processes with constraints (Glines, Biggs, and Bodily in press). The resulting model is called a constrained Hidden Markov process (CHiMP) which is visualized in Figure 2. The CHiMP model was chosen under the hypothesis that increased abstraction, resulting in increased generalization, would lead to a significantly larger solution space.

In implementing a filtration system, it is apparent that a large solution space is needed. Using two hypothetical models A and B (seen in Figure 3) we illustrate the restriction that solution space imposes on a system’s ability to step from a generalization system to a filtration system. Model A fails to have a solution space after filtering and thus remains a conceptualization for a generalization system. Model B, however, has a larger beginning solution space β due to an increase in the model’s ability to generalize the inspiring set.
Figure 2: A high-level schematic of a constrained hidden Markov process (CHiMP) of length 4 constrained so that the last word is “red” and the first word rhymes with “red”. Each column represents a position in the sequence to be generated. Each node represents a hidden state (i.e., part-of-speech) and a probability distribution for the observed states (i.e., words) that can be generated from that hidden state. By pruning observed states that are disallowed by constraints and then adjusting probabilities to maintain arc-consistency, the resulting model generates constraint-satisfying solutions with probability relative to the original probability distribution (Glines, Biggs, and Bodily in press). Hidden states pruned directly from applying constraints are indicated by dark grey nodes and states pruned during arc-consistency are indicated by light grey nodes.
Thus model $B$ has a usable solution space $\beta'$ after filtering and can be categorized as a filtration system.

**Results**

In demonstrating the increased generalization (and hence increased solution space) of CHiMP over CoMP, we compared the results of each model trained on the Corpus of Contemporary American English (COCA) (Davies 2009) and provided the same set of constraints. In particular, we selected training sets from the 2012 fiction portion of COCA and constrained each model to only output sequences in which the first letter of each word began with the same letter (e.g., a tongue-twister). We chose this problem because it represents a fairly general example of constrained sequence generation that is easily adapted to sequences of varying lengths.

Results are averaged over 26 instances of the problem with each instance having constraints defined with a different letter of the English alphabet.

Some qualitative results are shown in Figure 4. It should be noted that within the subset of 40 sequences generated by CHiMP, no duplicate or similar solutions where present; whereas 6 sequences were duplicates (or very similar) in the subset generated by CoMP.

We examined the effect of changing the sentence/model length on the novelty of the system in terms of the total number of unique solutions capable of being generated by each model (see Figure 5). As the sentence length increases, so too do the number of constraints on the sequence to be generated. In the abstracted CHiMP model, this is consequential; the model can afford to make restrictions at the observed node that do not affect transitions between sequence positions (which are isolated in the hidden layer). Only occasionally do a sufficient number of pruned states combine to require the pruning of a hidden state node, but such is a relatively rare occurrence.

By contrast, the effects of increased sentence length on the CoMP model are severely limiting. Each added position would typically add a number of novel unique solutions if it did not come with the addition of a new constraint. The newly constrained position has direct influence on previous observed sequence states and thus pruning values from the domain of these variables directly results in the removal of transitions between adjacent sequence positions. This results in a relatively slow growth in the solution space as sentence length grows.

The increase in the CHiMP model appears to be exponential owing to the multiplicative effect achieved by maintaining large domains for adjacent variables in the hidden layer.

Similar trends in the impact on novelty are manifest when we vary the training set size, keeping sentence length constant (see Figure 6). We see that the size of the solution space for the CHiMP model increases exponentially. The CoMP model also appears to have some slightly exponential growth, but at a significantly lower rate. This is again what we would expect to see. Increasing the training set size (when such is a possibility) still has a more significant impact on CHiMP than on CoMP model.

The results shown in Figures 5 and 6 suggest that CHiMP, with respect to CoMP, facilitates exponentially more novelty. The solution space of the CoMP model is by definition a subset of the solution space of the CHiMP model, and for most training and constraint sets will be a substantially smaller subset. It is expected that of the novel results produced by CHiMP, some will have higher value than the solutions shared by both models. Because the CHiMP model abstracts to a more significant degree from the training set than the CoMP model, we might expect a greater portion of the novel solutions to be of lower value. The suggestion from qualitative results shown in Figure 4 is that there is no obvious degradation of value. However, we do not currently have results to fully assess the extent to which value degrades (or doesn’t). In any case the expressivity of the CHiMP model enables a simple solution: introduce new or stricter filtering by increasing the number and stringency of constraints.

**Discussion and Conclusion**

In progressing from a generalizing system to a filtration system, our results provide meaningful insight into two important questions relating to Ventura’s spectrum of creative systems:

1. **For the filtration level of the spectrum, what challenges are CC systems likely to encounter?**

2. **What suggestions can be made to overcome these challenges?**

A significant challenge for CC systems attempting to transition to a filtration system is as more constraints (or filters) are put on the system, the solution space diminishes to the point of being too small to filter. As demonstrated in the CoMP model (Figure 5), the insufficient solution space prevents being able to apply more constraints and filters to produce higher quality artefacts.

The problem is not specific to our results or to Markov models. Filtering, by nature, reduces the solution space. As shown in Figure 3, any CC system with low generalization may fail to have a usable solution space after filtering.

Greater generalization can address the aforementioned problem. We see from our results that our model with greater generalization, CHiMP, excels in solution space size even as constraints are added (see Figure 5). The primary difference between CoMP and CHiMP is an added layer of abstraction in CHiMP that affords greater generalization. The solution to a diminished solution space is to increase the level of abstraction in the model. This increases the generalization ability of the model and results in a solution space substantial enough to “survive” filtering.

Increased constraints allow for greater creativity and quality because the system can use constraints to explicitly articulate and enforce the system’s goals and intentions. For example, in Markov models, increasing the Markov order (a form of adding more constraints) significantly improves the coherency of natural language, but the solution space is heavily diminished. With the CHiMP model, the solution space is sufficiently enlarged to avoid these devastating consequences to the solution space. Besides changes to the Markov order, other possibilities open up for using constraints to filter results to further improve quality, including...
Figure 3: The application of filters on two hypothetical models (A and B) demonstrates the requirement for larger solution spaces (increased generalization) in order to endure filtering with a usable solution space. Model B has a usable solution space after filtering; thus the model has moved further along in the spectrum from generalization to filtration.

**CoMP Tongue Twisters:**
- late last light levels like lady
- Diaz did dinosaurs died dell drove
- max moved my mother made my language lessons last look little lamb

**CHiMP Tongue Twisters:**
- queen Quanhe quite quiet queasy qualified
- flower facing forward for from forester
- free feeling facing followed free fate
- every educated Elizabeth expected Erika enchanting

Figure 4: Example results from generating 6-length tongue twisters (i.e., alliterative constraints) from both the CoMP and CHiMP models. Both models were trained on 10K sentences. Results are chosen from a randomly selected subset of 40 sequences from each model. The quality of tongue twisters is roughly equivalent between both models (both poor), but the CHiMP model is capable of generating exponentially more solutions. This suggests that increasing the Markov order in the CHiMP model (as an example of more stringent constraints) will have far less deleterious affects on the solution space as compared to a similar increase in the CoMP model.

Figure 5: The effects of sequence length on the number of total solutions generated by each model with a fixed training set size of 300 sentences. Both models are constrained such that each word in a sequence starts with the same letter; counts of total solutions are averaged over 26 runs (each run using a different letter from the English alphabet). We see that as the sequence length increases, total solutions for the CHiMP model increases exponentially (given the logarithmic scale) whereas the CoMP model stagnates.
Figure 6: The effects of training corpus size (number of training sentences) on the number of total solutions generated by each model with a fixed sequence length of 3. Both models are constrained such that each word in a sequence starts with the same letter; counts of total solutions are averaged over 26 runs (each run using a different letter from the English alphabet). The total solutions of both models increase in an almost parallel way; however, at 10K training sentences, CHiMP well exceeds 100M total solutions which contrasts CoMP at 1000 total solutions.

It is important to acknowledge the negative consequences of increasing the generalization in a learning model. In particular, generalization decouples dependencies between variables which can result in a loss of information during variable assignment. For example, generalizing to a hidden Markov model takes a significant toll on language coherence. In short, the novelty achieved by generalization comes with a trade-off in value. We hypothesize that this deterioration can be offset in the application of filters to preserve the information lost. We plan to examine this issue in future work.

Through developing a system (CHiMP) that more effectively achieves filtration, we have discovered insights into the challenges present in the leap from generalization to filtration and how to overcome them. The challenge of diminishing solution spaces can be overcome by amplifying the generalizing ability of the system through abstraction. Having realized the leap from generalization to filtration, the community is now poised to address the challenge of making the subsequent leaps along Ventura’s spectrum of creative systems, advancing past filtration into inception and ultimately creation.

References


Action Selection in the Creative Systems Framework

Simo Linkola¹, Christian Guckelsberger²,³,⁴ and Anna Kantosalo¹,²

¹Department of Computer Science, University of Helsinki, Helsinki, Finland
²Department of Computer Science, Aalto University, Espoo, Finland
³Finnish Center for Artificial Intelligence (FCAI)
⁴Game AI Research Group, Queen Mary University of London, London, UK
simo.linkola@helsinki.fi, christian.guckelsberger@aalto.fi, anna.kantosalo@aalto.fi

Abstract

The Creative Systems Framework (CSF) formalises creativity as search through a space of concepts. As a formal account of Margaret Boden’s descriptive hierarchy of creativity, it is at the basis of multiple studies dealing with diverse aspects of Computational Creativity (CC) systems. However, the CSF at present neither formalises action nor action selection during search, limiting its use in analysing creative processes. We extend the CSF by explicitly modelling these missing components in the search space traversal function. We furthermore integrate the distinction between a concept and an artefact, and provide stopping criteria for creative search. Our extension, the Creative Action Selection Framework (CASF), is informed by previous studies in CC and draws on concepts from Markov Decision Processes (MDPs). It allows us to describe a creative system as an agent selecting actions based on the value, validity and novelty of concepts and artefacts. The CASF brings more analytical depth for creative systems that can be modelled as utilising an action selection procedure.

Introduction

The process by which a creative product or artefact comes into existence represents one of the four central perspectives on creativity (Jordanous, 2016). For many Computational Creativity (CC) systems, this process can be described as ongoing selection and execution of actions through an agent. An action selection function hereby specifies how the agent chooses what to do next based on the current situation and its goals. Examples include a music robot’s selection of musical gestures (Hoffman and Weinberg, 2010), a game character’s next move (Guckelsberger, Salge, and Togelius, 2018) and a co-creative agent’s choice of collaboration partners (Hantula and Linkola, 2018). Vice versa, deciding on how an agent must select its actions to be creative can mark an important step in the design of a CC system.

Despite the salience of action selection in both the analysis and design of CC systems, there has been little previous effort to include action and action selection in a specialised formal framework for creativity. Existing research on creative agents either applies heuristic action selection (e.g. Saunders, 2012; Gabora and Tseng, 2017) or considers creative action selection in more general frameworks, such as Reinforcement Learning (RL) modelled on Markov Decision Processes (MDPs) (Vigorito and Barto, 2008; Colin et al., 2016). While these approaches offer sufficient solutions for their individual purposes, they are unfitted for analysing creative processes at large: the heuristic methods miss a unified formal foundation for comparison, and generic frameworks are not sensitive to the specifics of creativity, limiting the potential for the in-depth analysis of creative processes.

In this paper, we introduce the Creative Action Selection Framework (CASF), an extension to the Creative Systems Framework (CASF) (Wiggins, 2006a,b) which can serve as the formal foundation for describing and analysing action selection in individual creative agents. It is sufficiently general to be applied to different kinds of agents, from reflex-based to learning agents (cf. Russell and Norvig, 2009).

The CSF is a formal, mathematical account of Boden’s (2004) descriptive hierarchy of creativity. It defines Boden’s notion of a conceptual space by rules for valid concepts. These are coupled with rules for evaluation applicable to concepts within and out of the conceptual space. In the centre of the search process is a traversal function; based on rules for traversal, it allows the system to move from concept to concept. The interplay between these elements can be used to define characterisations of creative search which, in turn, can be used in analysing creative systems.

The CSF however misses an account of how this search is realised through an agent’s actions, and implicitly the selection of these actions. It hence has several limitations with respect to analysing creative agents: (1) it treats the traversal function as a black box and does not elaborate on how the agent decides which concept to move to next; (2) it does not distinguish between the concept, an agent’s inner representation of an idea, and the artefact, the concept’s external materialised expression (cf. Grace and Maher, 2015; Ventura, 2017); finally, (3) the CSF does not put forward stopping criteria, formalising how the agent reasons that the given concept and artefact are creative enough.

We extend the CSF to overcome these limitations, drawing on concepts in MDPs (Puterman, 2014). We (1) distinguish concepts and artefacts in the CSF. We (2) disassemble the CSF’s traversal function into constituents relevant for creative agents, allowing for action selection based on the value, validity and novelty of concepts and artefacts. We (3) describe possible stopping criteria for the search of concept-artefact pairs. We reflect on the lack of universal optimality criteria for creativity, which distinguishes the CASF action selection further from optimal MDP policies.
Background and Motivation
We first introduce the CSF, identify its limitations towards analysing creative agents, and introduce MDPs as inspiration to overcoming these limitations in the CASF.

The Creative Systems Framework
The CSF (Wiggins, 2006a,b) formalises Boden’s (2004) descriptive hierarchy of creativity. It hence allows for the abstract discussion of creative systems and the identification of relevant phenomena within, which particularly concerns the mechanisms of exploratory and transformational creativity. The CSF forms the basis for several studies in CC (e.g. Grace and Maher, 2015; Kantosalo and Toivonen, 2016; Alvarado and Wiggins, 2018; Linkola and Kantosalo, 2019). In this paper we focus on the exploratory part of the CSF, but the elements from our extension can also be used to drive the transformations of the system’s creative behaviour.

Exploratory creativity consists of discovering novel and valuable concepts within a known conceptual space (Boden, 2004). The CSF defines it as a septuple

\[ \langle \mathcal{U}, \mathcal{L}, [], \langle \ldots, \ldots \rangle, \mathcal{R}, \mathcal{T}, \mathcal{E} \rangle, \tag{1} \]

where the individual elements of the septuple are described in Table 1. Below, we only discuss the elements which are pivotal for the rest of the paper.

The universe \( \mathcal{U} \) is a multidimensional (possibly infinite-dimensional) space capable of representing anything. All possible distinct concepts \( c \in \mathcal{U} \) are distinct points in \( \mathcal{U} \). The empty concept \( \top \) is also part of the universe, \( \top \in \mathcal{U} \).

A function generator \( [\cdot] \) interprets a given rule set expressed in language \( \mathcal{L} \) and outputs a function which maps elements of the universe \( \mathcal{U} \) to real numbers in \([0, 1]\). It is used to generate functions that encode the rule sets \( \mathcal{R} \) and \( \mathcal{E} \).

The rule set \( \mathcal{R} \subset \mathcal{L} \) defines what kind of concepts are accepted as valid in terms of belonging to a certain class of objects such as a mathematical theorems or buildings in a specific architecture style. \( \mathcal{R} \) can be used to define the conceptual space, which Boden (2004) characterises as a structured style of thought, \( \mathcal{C} = \{ c \in \mathcal{U} \mid [\mathcal{R}](c) \geq k \} \), where \( k \in [0, 1] \) is a validity threshold.

The rule set \( \mathcal{E} \subset \mathcal{L} \) defines the evaluation function for the system. I.e. the function generated by \( [\cdot] \) through interpreting \( \mathcal{E} \) allows to evaluate any concept \( \forall c \in \mathcal{U} \) as \( [\mathcal{E}](c) \in [0, 1] \). We define the set of valued concepts in the universe using a value threshold \( l \in [0, 1] \): \( \{ c \in \mathcal{U} \mid [\mathcal{E}](c) \geq l \} \).

The system’s traversal of the conceptual space rests on a second function generator \( \langle \ldots, \ldots \rangle \). It takes into account the traversal rule set \( \mathcal{T} \subset \mathcal{L} \), specifying how the system moves from concepts (or sequences of concepts) to other concepts (or sequences). Since traversal can also be informed by \( \mathcal{E} \) and \( \mathcal{R} \), the generator interprets all three rule sets, \( \mathcal{T}, \mathcal{R} \) and \( \mathcal{E} \) into a function which maps a sequence of input concepts, \( c_{\text{in}} \), into a sequence of output concepts, \( c_{\text{out}} \):

\[ c_{\text{out}} = \langle \langle \mathcal{T}, \mathcal{R}, \mathcal{E} \rangle \rangle (c_{\text{in}}). \tag{2} \]

The CSF has been developed to, amongst others, describe and analyse the exploratory capabilities of creative systems. However, lacking the notion of actions, the CSF omits the decisions the system makes as the search unfolds. As such, it fails to describe creative agents and their behaviour in sufficient detail. Below, we discuss these limitations and draw on concepts from MDPs to address them.

Creative Agents and the CSF
We shape our concept of creative agents by combining Russell and Norvig’s (2009) concept of intelligent agents with the ‘standard definition of creativity’ (Runco and Jaeger, 2012). We consider a creative agent to be utility-based with the goal to produce creative, i.e. novel and valuable, concepts and artefacts. The latter distinction stems from separating an agent and its environment. A concept describes an agents’ inner representation of ideas. An artefact in contrast is a materialisation of a concept as part of the agent’s environment, which the agent may only have partial access to and control of\(^1\). The same concept can be expressed as different artefacts using diverse skills or means: the concept of a flying horse can be expressed as a poem or as a painting, and in both domains there are a plethora of ways to do so\(^2\).

During a creative agent’s operation, the system’s overall state can be conceived as a tuple of two states: one for the concept and one for the artefact, both of which can be “empty” at any given time. We denote this as the agent’s position. To produce concepts and artefacts, the agent performs actions that affect the state of its concept, the artefact’s, or both. The search for concepts happens in the concept state space, whereas artefacts are manipulated in the artefact state space\(^3\). The effectiveness of actions to induce change depends on the dynamics of these spaces.

We define the agent’s overall goal as finding a position where both the present concept and the artefact are assessed favourably, and the artefact is a good fit for the concept, i.e.

\( \mathcal{U} \) the universe containing all possible concepts

\( \mathcal{L} \) a language in which to express concepts and rules, in a broad sense of the universe, \( \mathcal{L} \subset \mathcal{U} \)

\( [] \) a function generator which maps a subset of \( \mathcal{L} \) to a function which associates elements of \( \mathcal{U} \) with a real number in \([0, 1]\).

\( \langle \ldots, \ldots \rangle \) a function generator mapping three subsets of \( \mathcal{L} \) to a function that generates a new sequence of concepts of \( \mathcal{U} \) from an existing one.

\( \mathcal{R} \subset \mathcal{L} \) rules defining valid concepts

\( \mathcal{T} \subset \mathcal{L} \) rules defining traversal in the concept space

\( \mathcal{E} \subset \mathcal{L} \) rules defining evaluation of concepts

Table 1: Description of the elements in the CSF

---

\(^1\)This environment can be arbitrarily complex. Here, we only consider the part constituting the agent’s artefact.

\(^2\)Our concept-artefact distinction is informed by Grace and Maher (2015) and Ventura’s (2017) genotype-phenotype distinction.

\(^3\)We acknowledge that this distinction of an internal conceptual space and an external environment with artefacts is a simplification as from a monist position (Jaworski, 2011), there is no difference between these domains in terms of substance. Moreover, the existence of a clear boundary separating an agent and its external environment is disputed, challenging the very concept of agency.
it expresses the concept well. Crucially, the agent may first
thrive to find a prominent concept and then express it as
an artefact. Alternatively, it may seek an artefact which is
then associated with a concept, as in the case of Duchamp’s
“ready-mades” where an existing artefact is placed in a new
context in which it can be conceived in a fundamentally dif-
ferent way. Moreover, the agent may alternate between con-
cept and artefact exploration, incorporating insights from
perceiving the unfinished artefact along the way.

In existing CC systems, the exploration of potential con-
cepts and artefacts is often interrupted externally by the user
or designer as soon as they are satisfied by the outcome. We
however model a creative agent’s own perspective on its con-
cept and artefacts. Hence, we must formalise how the agent
itself decides that it should stop its creative process.

We identify several shortcomings of the CSF towards de-
scribing creative agents as introduced above:

1. No actions: The agent’s actions are obscure in the traversal
function and the CSF lacks a formal account of the agent’s action selection in the creative process.

2. No concept-artefact separation: The CSF does not distin-
guish between concepts and artefacts and hence cannot explicate how their possible relationships can impact the agent’s creative process.

3. No stopping criteria: The CSF offers little explanation of when a creative agent should “stop” its exploration, e.g., to start anew or to output its current concept-artefact pair.

We address these limitations by drawing inspiration from the framework of Markov Decision Processes.

**Markov Decision Processes**

A Markov Decision Process (MDP) (Bellman, 1957) describes a sequential decision-making problem. It models the possible interaction between an arbitrary agent and its environment over time, where the environment is distinguished from the agent by everything that they cannot change arbitrarily (Sutton and Barto, 2018, p. 50). At each point of the interaction, the agent can receive a reward from the environment. The problem consists of finding a policy, i.e., a decision-making rule, which maximises cumulative future reward from an initial state onward (Puterman, 2014, p. 2).

A (infinite horizon) MDP is a quadruple $\langle S, A, p, \rho \rangle$ (Puterman, 2014, pp. 1-2). The environment state at time-step $t \geq 0$ is denoted as $s_t \in S$. An agent’s action $a_t \in A$ can influence the future state of the environment $s_{t+1}$ determined by environment dynamics $p : S \times A \times S \to [0, 1]$ given as a conditional probability distribution $p(s_{t+1} | s_t, a_t)$, where the actions available to an agent can depend on the current environment state. The actions produce an immediate reward signal $r_{t+1} \in \mathbb{R}$ determined by the reward function $\rho : S \times A \times S \to \mathbb{R}$, given by $\rho(s_{t+1}, a_t, s_t) = r_{t+1}$. The Markov assumption implies that $s_t$ must encode all relevant information about the past agent-environment interaction that matters for the future dynamics.

All previous elements formalise the decision-making problem, but do not provide a solution. Specifying the probability of the agent selecting a specific action in a specific state $\pi(a | s)$, a policy is an attempt to solve the problem specified by a MDP. A solution to the problem is an optimal policy $\pi^*$ that maximises cumulative future reward. Crucially, there can be several optimal policies.

MDPs and methods to (approximately) solve them, e.g., RL, have been previously discussed and utilised in creative contexts (e.g., Hoffman and Weinberg, 2010; Han- tula and Linkola, 2018). Vigorito and Barto (2008) model creativity as a blind variation-and-selection process, arguing that creative behaviour may be acquired by hierarchical RL as means to reduce the complexity of creative search. Colin et al. (2016) present one possible mapping of Ritchie’s (2012) “simplified version” of the CSF to hierarchical RL where the MDP policies are CSF’s concepts and their evaluation in the CSF is likened to the discounted return of the policy. Based on this mapping, they argue that hierarchical RL realises creative behaviour in the form of exploratory and transformational creativity.

We next move beyond existing work by proposing an extension to the CSF which accounts for actions and action selection by adopting the notions of states, actions and environment dynamics from MDPs. We also elaborate on the use of these notions to fit the specifics of creative agents, in particular the distinction between concepts and artefacts.

**Bringing Actions to the CSF**

We now introduce the Creative Action Selection Framework (CASF) as extension to the CSF to leverage its power for the description of creative agents. We consider a single creative agent with a closed loop creative process. That is, the agent moves in the search space using solely its own actions and reasoning. The agent’s actions can change the states of the concept and artefact spaces, which the agent can then observe and assess. We thus do not explicitly consider any co-creative and interactive agents which rely on external feedback or communication with other agents. However, most of our formulation also applies to these cases.

Our main contribution in this section is the deconstruction of the agent’s traversal function $\langle T, R, E \rangle$. Before we can address this however, we must add detail to some of the CSF’s other elements: we introduce the concept of actions into the traversal rules $T$, and make time and the concept-artefact distinction explicit in the framework.

**Universe $\mathcal{U}$:** We distinguish two subsets from the universe: $\mathcal{U}_c$, which encompasses concepts, and $\mathcal{U}_a$, which contains artefacts. $\mathcal{U}_c$ and $\mathcal{U}_a$ comprise the possible states of concept and artefact search, respectively. The empty state $\emptyset$ is a member of both subsets.

**Input and output sequences $c_{in}$ and $c_{out}$:** As we view a creative agent in a combination position of a concept and an artefact, we consider each element in the input and output sequences $c_{in}$ and $c_{out}$ as a position tuple $\phi = (\omega, \alpha)$, where $\omega \in \mathcal{U}_c$ and $\alpha \in \mathcal{U}_a$. Thus, for a single element sequence, an agent’s traversal (Equation 2) may be denoted by

$$\phi_{out} = \langle T, R, E \rangle(\phi_{in}).$$

(3)

In the rest of this paper, we mostly consider such singular input and output positions. However, as $\mathcal{U}$ can represent
anything, it can also comprise sequences of such positions. Our formalisation works mostly for both cases.

Time $t \geq 0$: Time is implicitly present in the original CSF’s ordering of input and output sequences. We make it explicit in an agent’s closed loop creative process by denoting $\phi_{\text{in}} = \phi^t$ and $\phi_{\text{out}} = \phi^{t+1}$. That is, we assume that time progresses with traversal from $t$ to $t + 1$, and that the output of the last time step’s traversal function is fed as an input to the traversal function in the next time step.

**Value, validity and novelty:** Novelty is not explicitly considered in the original CSF – what the evaluation rules $E$ entail is left vague. To make novelty explicit, we denote the rules for novelty by $E_N^+$, and rules for evaluation by $E_E$. Furthermore, we modify both the interpreted validity and evaluation function ($[R]$ and $[E_E]$) to accept a position as input and address novelty by adding an interpreted novelty function $E_N^+$ which also operates on a position. We assume that these functions return a triplet of real values representing the assessment of the input concept, the artefact and their combination. This results in the following functions:

\[
\text{evaluate}(\phi) = [E_E](\phi) = (e_\omega, e_\alpha, e_{\omega \alpha}) \quad (4)
\]

\[
\text{validate}(\phi) = [R](\phi) = (v_\omega, v_\alpha, v_{\omega \alpha}), \quad (5)
\]

\[
\text{novelty}(\phi) = [E_N^+](\phi) = (n_\omega, n_\alpha, n_{\omega \alpha}), \quad (6)
\]

where the subscript $\omega$ denotes assessment for the concept, the subscript $\alpha$ assessment for the artefact, and the subscript $\omega \alpha$ assessment for the combination$^4$.

The combination assessments describe if the artefact expresses the concept properly (validity) and how elegant the expression is (evaluation). For example, for the concept of freedom, valid artefacts could portray unlocked shackles or a bird. However, if the bird happened to be a penguin, the concept might not be as elegantly expressed as using a more stereotypical bird in a proper context.

For novelty, as $\phi$ is a single position for which novelty is computed, we assume that the function incorporates the selected input positions (such as those that the agent chooses to output) into a persistent, internal model of novelty for successive calls. That is, the novelty is computed with respect to the history of the agent. Novelty of the combination serves here a similar purpose as the combination assessments in validity and evaluation functions: it may be used to assess how new the concept-artefact pair is.

**Traversal rules $T$:** Traversal rules govern how the system explores the search space. As we model the agent’s exploration as an action selection process inspired by MDPs, we distinguish the following subsets in $T$:

$T_A$ (or simply $A$): the actions available to the agent;

$T_D$: the agent’s model of the search space (how the concept and artefact spaces react to the agent’s actions); and

$T_C$: the policy specifying the habit to move in both spaces.

$^4$Specific traversal rules can foster novelty, but it is ultimately the result of evaluation. In contrast to Grace and Maher (2015), we hence only consider it in evaluation and not in the traversal rules.

$^5$Further distinctions are possible, e.g. the evaluation of the artefact given the concept, $e_\alpha|\omega$, and vice versa, $e_\omega|\alpha$.

---

![Figure 1: Closed loop traversal: an agent (1) observes its position, (2) selects the next action and (3) executes it, which may influence the position. Different action selection mechanisms use some or all of the following components in varying order: the assessment of the present position, the filtering of actions into a relevant subset, the prediction of action outcomes, and the choosing of an action.](image)

The actions $\alpha \in A$ form the nucleus of action selection and of our deconstruction of the traversal function. An action may alter the agent’s place in the concept space, in the artefact space, or in both. By construction, CASF also includes two special action types present in creative processes and previously discriminated in CC, translation and re-perception (Grace and Maher, 2015; Ventura, 2017). The agent is completely free to execute which action it chooses, but the actions available to the agent may differ from position to position.

**Deconstructing the Traversal Function**

Next, we go through a single iteration of the agent’s closed loop traversal encompassed in the traversal function $\langle T, R, E \rangle$, and illustrated in Figure 1. The agent initially observes its current position, and eventually executes the next action. The action selection can be implemented via different mechanisms; each uses some or all of the following components, potentially in different order: the agent assessing the position’s value, validity and novelty, filtering its own action possibilities, predicting their possible outcomes, and choosing the next action to take. Below, we address these six components individually and separate them into their own functions. This provides us with conceptual clarity, and supports the CASF’s use in the description and analysis of diverse creative agent types, including hard-coded reflex-based agents and those capable of learning from experience to adapt action selection. We acknowledge that this separation does not account for all atomic elements in creative action selection – e.g. predicting could be further divided into predicting the next positions and assessing them.

**Observe (position):** The agent needs to observe its position at time $t$ to gather information about where it is in the concept and artefact spaces. Formally,

\[
\phi_{\text{obs}} := \text{observe}(\phi^t), \quad (7)
\]

where $\phi^t$ is the agent’s actual position at time $t$ and $\phi_{\text{obs}} =$
\( (\omega^t, \alpha^t), \omega^t \in \mathcal{U}_\omega \) and \( \alpha^t \in \mathcal{U}_\alpha \), denotes its observation by the agent. Generally, this observation is imperfect, i.e. \( \phi^t \neq \phi_{\text{obs}} \), the agent may not have a full view of the universe, and observe some of the dimensions of \( \mathcal{U} \) (relevant for either concepts or artefacts or both) incorrectly or not at all. In the rest of this section, we typically assume that \( \phi_{\text{obs}} \) is a single position. However, whenever we deal with mechanisms which include the agent’s history, we assume that \( \phi_{\text{obs}} \) contains the agent’s (recent) observed position history or the agent has other means to retrieve it from data structures which are not explicitly specified.

**Assess (position):** The agent may assess its observed position. We denote this assessment by

\[
\delta := \text{assess}(\phi_{\text{obs}}, R, E) = \left[ \begin{array}{c} E_E(\phi_{\text{obs}}) \\ R(\phi_{\text{obs}}) \\ E_N(\phi_{\text{obs}}) \end{array} \right]^T, \tag{8}
\]

where the three functions on the right are defined in Equations 4-6. These assessments are essential to direct the agent’s further reasoning process. For example, if all assessments are favourable, the agent may choose to output the concept-artefact pair; if only the concept is assessed favourably, the agent may only develop the artefact further.

**Filter (actions):** The agent may filter which actions are appropriate given its current position and the assessments:

\[
\mathcal{A}_{\text{app}} := \text{filter}(\phi_{\text{obs}}, \Delta, T_A), \tag{9}
\]

where \( \mathcal{A}_{\text{app}} \subset T_A \) is the set of actions the agent finds appropriate with respect to its goal(s), and \( \Delta \) is either the returned assessments, \( \delta \), or (parts of) the assess-function itself.

The filtering step adheres to the “hard traversal rules” of the system which can not be broken in any circumstance, but it may also be based on the agent’s (recent) history and goals. Typical cases of the former are the hard-coded restrictions for the system to stay in the conceptual space, e.g., by restricting the percentage of the canvas that can be red. The latter part can serve similar purposes as “focus” or “attention” in animals. If the agent’s current goal e.g. is to compose a painting using only triangles, then the appropriate actions in the artefact space may deal only with triangles.

**Predict (actions):** The agent may inform its next action by predicting the consequences of available, potentially filtered actions. This may involve a (learned) model of action outcomes, or some fixed heuristics. Below, we assume the former case, where the agent uses its model of the exploration dynamics from MDPs. Formally, these dynamics \( T_\phi : \mathcal{U}_\omega \times \mathcal{A} \times \mathcal{U}_\omega \rightarrow [0, 1] \) are given by a conditional probability distribution \( T_\phi(\omega^{t+1} | \omega^t, \alpha^t, \alpha^t) \). The predict function for an arbitrary action is given by

\[
(\hat{\phi}^{t+1}, \hat{e}^{t+1}, \hat{v}^{t+1}, \hat{p}^{t+1}) := \text{predict}(\phi_{\text{obs}}, a, R, E, T_\phi), \tag{10}
\]

where \( \hat{\phi}^{t+1}, \hat{e}^{t+1}, \hat{v}^{t+1} \), and \( \hat{p}^{t+1} \) are the predicted observed position in the next time step caused by the action and its predicted value, validity and novelty triplets, respectively. In general, the output\(^6\) can also be a sequence of \((\hat{\phi}^{t+1}, \hat{e}^{t+1}, \hat{v}^{t+1}, \hat{p}^{t+1}, \hat{p})\) tuples, where \( \hat{p} \) is the predicted probability for that outcome. We have these variables to indicate that they represent potential rather than actual states.

Where the filter-function restricted the appropriate action set with respect to the agent’s history and goals, the predict-function envisions likely outcomes of the actions and approximates their assessments. The predictions may be inaccurate and incomplete as the agent’s model of the exploration dynamics may only encode some of the relevant concept and artefact dimensions. Strong predictive capabilities allow the agent to assess an action more thoroughly by exploring its consequences further into the future. Informally, an agent employing a predictive model “imagines” possible concepts and artefacts before they are realised, and can use that imagination to drive its creative process.

Apt prediction capabilities are essential when executing an action (see below) is resource demanding or non-reversible. For example, a painting robot may take a considerable time in executing a set of instructions to paint the next patch and it costs money to buy the paint needed. Moreover, in safety critical domains, the actions may cause dangerous situations and ultimately harm humans or other actors.

**Choose (action):** The last part of the action selection is naturally about choosing the next action to take. This may be entirely random, but can also be informed by the current position’s assessment and by predictions of potentially filtered actions. Moreover, it could take into account \( T_\pi \), (learned) heuristics on how to proceed from this (or similar) observations onwards as adaptation of the MDP policy\(^7\). As a creative agent is constantly in a combination of two states, we have \( T_\pi : \mathcal{U}_\omega \times \mathcal{A} \rightarrow [0, 1] \), which is given as a conditional probability distribution \( T_\pi(a^t | \omega^t, \alpha^t) \).\(^8\)

The choose-function incorporates these information sources into a mapping to a single chosen action. Formally,

\[
a := \text{choose}(\phi_{\text{obs}}, \Delta, D_{\text{preds}}, T_\pi), \tag{11}
\]

where \( \Delta \) denotes either the returned assessments, \( \delta \), or (parts of) the assess-function itself, and \( D_{\text{preds}} \) is a potential mapping from actions to their predictions obtained using the predict-function. If the agent can only filter, the mapping contains only the keys for each \( a \in \mathcal{A}_{\text{app}} \).

Creative agents can implement the choose-function in different ways. It may use well known methods, e.g. a softmax (Sutton and Barto, 2018, p. 322) over action assessments, or it can incorporate pre-coded heuristics. Moreover, the function may depend on the recent history of the agent, e.g., to determine the priority of novelty, value and validity, or to compare the assessments of the current position and the predicted outcomes with past assessments.

**Execute (action):** Executing the chosen action is the agent’s means to potentially impact its position. Formally,

\[
\phi^{t+1} := \text{execute}(a), \tag{12}
\]

where \( \phi^{t+1} \) is the objective position at the next time step. This is the output of the traversal function.

---

\(^6\)The predict-function may also return confidences for the predictions, which are argued to be required for assessing surprise (Grace and Maher, 2015) of the action outcomes.

\(^7\)Considering the policy as argument to choose allows us to consider off-policy traversal and policy updates in the CASF.

\(^8\)This formalisation also accounts for deterministic policies by modelling them as Dirac delta distributions.
Algorithm 1 Example of model-free action selection in computing the traversal function $φ_{t+1} = ⟨⟨T,R,E⟩⟩(φ_t)$.

\begin{verbatim}
φ_{obs} ← observe(φ_t)
δ ← assess(φ_{obs}, R, E)
store (a^{t-1}, δ) in dictionary D_δ
a_t ← choose(., D_δ, .)
φ_{t+1} ← execute(a_t)
\end{verbatim}

Algorithm 2 Example of model-based action-selection in computing the traversal function $φ_{t+1} = ⟨⟨T,R,E⟩⟩(φ_t)$.

\begin{verbatim}
φ_{obs} ← observe(φ_t)
A_{appr} ← filter(φ_{obs}, .., T_A)
foreach a ∈ A_{appr} do
    (φ_{t+1}, \hat{e}_{t+1}, \hat{\delta}_{t+1}, \hat{n}_{t+1}) ← predict(φ_{obs}, a, R, E, T_φ)
    store (φ_{t+1}, \hat{e}_{t+1}, \hat{\delta}_{t+1}, \hat{n}_{t+1}) in dictionary D_{preds}
end
a_t ← choose(φ_{obs}, D_{preds}, T_π)
φ_{t+1} ← execute(a_t)
\end{verbatim}

The next traversal cycle starts with the agent observing its potentially new position as basis for action selection and to verify whether the previous action has resulted in a desirable or predicted outcome. This is not certain: the agent’s predictions may be defective, the dynamics of the performed action may be noisy, e.g. when splattering paint, or the execution may add imperfections to the planned action, e.g. by an unintentional rotation of a loose joint.

There exist many means to combine the above building blocks into a specific action selection function: many creative agents do not use all elements, or their functionality overlaps in the specific implementation. To illustrate the spectrum of possible approaches and support the applicability of the CASF across different agent types, we provide pseudocode for two action selection mechanisms in the traversal function. In the model-free approach in Algorithm 1, the agent chooses the next action to perform based on a past record of action-assessment tuples. A concrete example of this approach is Q-learning (Sutton and Barto, 2018, p. 131 ff.). In the model-based approach in Algorithm 2 in contrast, the agent predicts the consequences of each action in a set of previously filtered, appropriate actions. It then leverages these future action assessments to choose the next action to execute. An example for this is Monte Carlo Tree Search (Sutton and Barto, 2018, p. 185 ff.).

Stopping Exploration

By repeatedly invoking its traversal function based on the previous cycle’s output, the agent moves in the concept and artefact space. Crucially, the original CSF does not specify any stopping criteria for exploratory creativity to explain how the agent reasons that it has arrived at a particularly apt concept-artefact pair, which it could, for example, then show to others. Below, we describe a few potential stopping criteria, partly informed by the additional elements in the CASF:

**Thresholds:** A simple way for the agent to decide that the given concept-artefact pair is creative enough is that all the assessments are above some absolute thresholds given to the agent during its initialisation. This relates to Wiggins’s (2006a) usage of a filtering thresholds for the conceptual space and the set of valued artefacts. Dynamic thresholds work in a similar manner, but instead of the thresholds being fixed, the agent may alter them based on its own history and experience on assessing concepts, artefacts and their combinations. This gives the agent more room e.g. to determine acceptable assessments in a certain area of the search space.

**Predictions:** Given a predictive model, the agent may approximate if it could, by means of its acting, cause a concept-artefact pair in the near future that is assessed more favourably than the current one. If the likelihood for this is high, then the agent may reason that it should not stop in the current position. If the likelihood for generating a better position is low however, the agent may either output the position (if its overall assessment is favourable) or start anew (if the current position is below average). Predictive reasoning and stopping is especially important in creative domains where actions can hardly be reversed, e.g. when painting on a physical canvas, or in music and dance improvisation.

**Resource restrictions:** The agent may stop exploration based on the consumption of a specific, tracked resource. For instance, it may have a time budget, and stop as soon as additional exploration is not predicted to yield any improvements in assessing the current concept-artefact pair.

**Other criteria:** Naturally, there are a multitude of other possible stopping criteria for the agent based on its design and purpose. Moreover, the criteria above may be combined.

**Action Selection in Analysis**

Next, we consider how the CASF may be applied in describing and analysing creative agents. Due to the scope of the paper, we refrain from most mathematical formulations and merely list dimensions which may provide a useful starting point for an in-depth analysis of an agent’s creative process.

We first distinguish dimensions that enable, amongst others, the coarse analysis of a creative agent’s action capabilities: movement in the position space (Do the actions cover movement in both the concept and the artefact space?), action possibilities (How many actions are available to the agent in any given situation? How many of these result in distinct outcomes?), action granularity (How fine-grained are the agent’s potential movements in the concept and artefact spaces?), execution control (Does executing an action in a certain position map to one or to multiple outcomes?), action scope (Does the agent have actions that are fundamentally different, e.g. actions for producing both music and visual art?), and action learning (Is the agent able to learn new or more complex actions during its creative process?).

The dimensions above may serve as the basis of analysing a creative agent. However, they neither provide sufficient detail on the characteristics of the action selection procedure inside the traversal function, nor do they govern the agent’s overall process of moving in the position space (i.e. action-position sequences) while searching for apt concept-artefact pairs. We address (parts of) both of these cases below.

**Filtering characteristics:** Does the set of appropriate ac-
tions change between positions? How strict is the agent’s filtering? Too strict filtering may restrict creativity, while lenient filtering may hinder performance. Controlled oscillation between lenient and strict filtering may hint that the agent’s creative process can be characterised with cycles of divergence and convergence.

Prediction abilities: Is the agent able to predict outcomes of its actions? How well the agent’s prediction matches the outcome of the executed action? How reliable are the predicted value, validity and novelty? Prediction ability is important for the agent to work towards its goals, while mistakes in predictions may give arise to serendipity.

Flexibility of the process: Does the system’s operation result in similar action sequences or does the action sequences differ between producing multiple concept-artefact pairs for output. Is the process largely the same no matter which concept and/or artefact is produced? Overall flexibility of the process can be seen as desirable, but it should be coupled with the reliability of the process.

Reliability of the process: How reliably the creative agent produces apt concept-artefact pairs? How well the agent is able to exploit promising subspaces of the whole position space? This dimension analyses the agent’s alignment towards its overall goal. It should be used in conjunction with the other dimensions to ensure that the agent does not simply wander around.

Lastly, we reflect on the deterrent modes of creativity which deal with the system’s overall deficiencies to reach apt concepts (Wiggins, 2006a). With the CASF, we can describe these modes from the agent’s perspective. An agent which only momentarily visits any of these deterrent modes can be argued to be aligned towards its goal of producing apt concept-artefact pairs. However, some exploration in these modes may be needed in order for the agent to reach apt positions which would not be discovered otherwise.

In order for the original mode descriptions to be applicable for creative agents, we need to modify them (1) to take time into account and (2) to allow concept-artefact pairs. We use generative uninspiration (Wiggins, 2006a) as an example, but similar modifications can be done to all of the modes. For simplicity, we only cover the case of evaluating the concept-artefact combination, but any composition of concept and artefact assessments could be considered. Below, we use time leniency, denoted by m, to mark the agent’s own understanding of what is an acceptable number of time steps to continuously spend in a deterrent mode.

Definition. Generative Uninspiration Let \( \phi \) be an agent’s observation of its position in the time step \( t \) and let \( [E^t_{\text{c}}(\phi^t)] = (e^t_{\phi^t}, e^t_{\phi^t}, e^t_{\phi^t}) \) be the agent’s evaluations for its observed position in the time step \( t \). An agent with time leniency \( m \) notices itself exhibiting generative uninspiration (for concept-artefact pairs) on time step \( k \geq t + m \), if \( \forall e_{\phi^t} \in (e^t_{\phi^t}, \ldots, e^k_{\phi^t}) : e_{\phi^t} < \epsilon \), where \( \epsilon \) is the agent’s threshold for valuable concept-artefact combinations.

Discussion

We have defined a creative agent’s main goal as the production of creative, i.e. novel and valuable, concept-artefact pairs. We have moreover formalised the respective process as ongoing selection and execution of actions based on the assessment of these pairs, resulting in a sequence of action-position tuples. The CASF hence provides a deeper account of the product and process perspective on creativity (Jordanus, 2016) than the original CSF, in that it characterises the desired properties of creative products, and uses them to ground the creative process in action-selection. Our extension accounts for the person perspective by considering the extent of the agent’s creative capabilities in a similar but more detailed manner as the original CSF’s description of deterrent modes of creativity. For instance, the CASF allows us to ask if the agent has the ability to predict the consequences of its actions.

The concept-artefact distinction in the CASF is included for completeness, as it is present in theories of human creative processes and has been discriminated before in CC (Grace and Maher, 2015; Ventura, 2017). However, not all agents are capable of using both state spaces. In these cases, the framework can be reduced to deal only with concepts or artefacts.

Creative processes can also be considered through the FACE model (Colton, Charnley, and Pease, 2011), which aims to describe progress in the development of a creative system. The FACE model distinguishes between concepts (a piece of code, e.g. a function) and expressions generated using these concepts. This bears some similarity to concepts in the CASF and their expressions as artefacts. However, the FACE model handles the expression generation as a single act (similar to Ventura, 2017), whereas in the CASF an artefact may be generated using multiple actions. Moreover, a CASF agent may search for an apt concept for a fixed artefact, a case that is not accounted for in the FACE model.

In the CASF, the observe-function is treated as a black box. However, certain theories of creative processes, such as engagement-reflection (Sharples, 1996), treat the perception of an artefact as an action in its own right. While the CASF encompasses re-perception actions that describe an artefact as a concept, it does not explicate all the different types of perceiving actions, some of which may involve assessing the artefact in a particular way. We believe that the framework would benefit from making them explicit.

Surprise is a prominent subjective assessment which is sometimes considered a defining characteristic of creativity (Boden, 2004), and which may influence action-selection and hence the direction of a creative process. We have not considered surprise in this paper, but it could be included into the CASF by e.g. adopting Grace and Maher’s (2015) proposal: the predict-function can be modified to return confidence values for the predictions and the prediction accuracy can then be determined in the next time step. If the prediction confidence was high and the observed position or an assessment in the next time step deviates sufficiently from the prediction, the agent may quantify this as surprise.

Conclusions

We have extended the CSF to distinguish concepts and artefacts and to include action selection in its traversal function. The resulting Creative Action Selection Framework (CASF)
is the first formal framework oriented towards the analysis of creative agents. It provides formal tools to analyse a creative agent’s on-going process of producing concepts and artefacts by dividing the agent’s traversal of the search space into six conceptually separate building blocks which may each be further specified and analysed. By formalising the agent’s action and position sequences and their alignment with their goals, the CASF moreover also affords a more high-level analysis of the resulting behaviour.

One goal for future work is to detail the formalism further so that the components in an agent’s action selection can be considered in terms of simpler, shared atomic factors. This would highlight their commonalities and support the analysis and comparison of creative systems. By introducing action-selection to the CSF, the CASF allows us to compare a dedicated framework for the analysis of creative agents to more general decision-making frameworks in AI. Another goal for future work is to investigate such possible mappings and consequently evaluating the specific notions of creativity developed by Boden (2004) and formalised by Wiggins (2006a,b) in a wider AI context.

Acknowledgments

SL is funded by Academy of Finland grants 313973 (CACS), and 328729 (CADCAR). AK is funded by Academy of Finland grant 311090 (Digital Aura) and was funded by the doctoral programme in Computer Science at the University of Helsinki. CG is funded by the Academy of Finland Flagship programme “Finnish Center for Artificial Intelligence” (FCAI) and was funded by EPSRC grant [EP/L015846/1] (IGGI).

References


Humans in the Black Box: 
A New Paradigm for Evaluating the Design of Creative Systems 

Brad Spendlove and Dan Ventura 
Computer Science Department 
Brigham Young University 
Provo, UT 84602 USA 
brad.spendlove@byu.edu, ventura@cs.byu.edu

Abstract

Modular creative systems employ specialized modules that take on challenging responsibilities. Such modules can be expensive to develop and are often imperfect, leading to uncertainty in whether a system’s poor results are due to module imperfection or larger systemic issues. Computational creativity research would benefit from a method for validating the design of creative systems before expending development resources implementing them. We present ideal-module prototyping (IMP), a design validation paradigm that serves these needs by using modules that delegate responsibilities to humans as an ideal against which to compare system expectations and performance. We relate an experiment we conducted in applying IMP to an existing creative system to demonstrate the insights that can be gained from this method. We argue for the widespread adoption of modular creative system design validated by ideal-module prototyping.

Introduction

Computational creativity research has as one of its goals the invention of computational systems that can be said to exhibit creativity. This is a lofty and challenging goal that approaches the limits of humans’ understanding of our own mind and consciousness.

Given such a challenging goal, it is no wonder that successful computational systems that advance closer to that goal are complicated and at times difficult to understand themselves. Explainability is a hot-button topic in many stripes of AI research (Ehsan and Riedl 2020), as advanced machine learning algorithms increasingly exceed their own creators’ abilities to predict and control.

Computational creativity researchers, who create systems that by necessity are complex and unpredictable, often face similar challenges. The complexity of computationally creative systems, coupled with the vast spaces of possible outputs in a given creative medium, can result in systems that produce low-quality output but whose internal design flaws are difficult to diagnose.

One approach to solving these issues is to implement a prototyping method that will reveal system flaws before they result in low-quality output. Another is to develop tools for more nuanced diagnosis of existing systems with low-quality output. The development of a useful paradigm for prototyping the designs of new creative systems and diagnosing the flaws of existing systems would be beneficial to the computational creativity community.

Modular Creative Systems

Modular creative system designs use specialized modules that compute tasks—such as knowledge lookup, transformation, combination, and evaluation—connected by higher-level logic that organizes the flow of information between them. The tasks implemented by these modules often represent challenging operations that attempt to reach human-level performance. For example, a module could attempt to represent knowledge similarly to a human or evaluate a creative artifact as a human would. Even tasks like computer vision that do not seek to produce the same output as a human still often need to function on a level of performance equal to that of a human.

Because these tasks are extremely challenging, the modules that implement them commonly represent the weak links in a creative computational system. Conversely, if such a module is working as intended, but the system as a whole still produces low-quality results, it is important to make that distinction and not expend effort on modifying a module that already functions well.

With a number of interfering parts, many of which are attempting to complete such challenging tasks, it can be difficult to diagnose problems with a modular creative system as a whole. When the quality of the system’s output is low, what is the cause? The ability to tease apart the interplay between modules and higher-level control would be useful both to reduce diagnostic complexity and to analyze tricky modules without conflating issues from other modules or information flow.

Ideal Modules

One approach to evaluating a modular creative system is to compare its performance to a version of that system with ideal modules. Each module that is replaced with an ideal version is one that cannot be the source of flaws in the system’s output. If all of the system’s modules are ideal, then any flaws in its output must be the result of how those modules are interconnected or some other holistic aspect of the system design. Alternatively, if the system yields high-quality output with ideal modules, then any reduction in out-
put quality when the same system uses imperfect modules can be attributed to the difference in module quality.

For some tasks such as mathematical operations, existing computational modules represent the ideal. However, many tasks that are useful in computational creativity do not currently have ideal computational implementations. We propose that for these types of modules, a useful definition of an ideal module is one that delegates completion of the task to a human. This conception of ideality serves creative systems whose modules or output are intended to mirror human performance, be compatible with a human audience, or achieve human-level quality.

We will refer to such modules as human-delegated modules, to indicate that they complete tasks by presenting the task to a human to solve. The inclusion of human-delegated modules in a creative system would obviously discount it from being considered an autonomous computationally creative system. However, we propose that testing a creative system using ideal modules, even if they are not computational, is a powerful tool for prototyping and evaluating the algorithmic validity of the system’s design. Once the design has been validated, effort can be expended to engineer a purely computational implementation of the system.

Notably, this comparison can be usefully made at the design stage of creative system development, before developing complex computational modules. By first prototyping the system with ideal, human-delegated modules, its designers can evaluate the performance potential of its high-level design before investing effort in the expensive process of developing a fully computational system.

This new paradigm, which we call ideal-module prototyping or IMP, represents a powerful tool that is not currently available to the computational creativity research community. It gives researchers a method for validating the designs of new creative systems and for diagnosing flaws in existing creative systems by comparing them to an idealized version.

In this paper, we argue for the following position: computational creativity research should be conducted using modular system design evaluated via ideal-module prototyping. We describe ideal-module prototyping in detail, relate an experiment we conducted with applying IMP to an existing creative system, present arguments to defend our position, and discuss the impact that this paradigm could have on the future of computational creativity research.

**Ideal-module Prototyping (IMP)**

This prototyping paradigm is powerful but not overly complex. The main insight is that much can be learned about a modular creative system by replacing its computational modules with human-delegated versions that fulfill the same task or operation. In this section we detail the paradigm, and in the next we relate an experiment we conducted with it.

Note that although IMP is primarily intended for prototyping the design of a new creative system before the system itself is engineered, it can also be retroactively applied to evaluate and diagnose completed systems. The paradigm is applied similarly in either case, with the main difference being that the computer modules of an existing creative system are already well-defined and can be compared to directly.

**Building Human-delegated Modules**

When the goal of a creative system is to produce human-compatible output, the generation of that output often requires knowledge or computation that mimics human cognition to some degree. Modular creative systems have specialized modules that are called upon to perform these critical tasks in the creative process. Because these tasks attempt to operate at a human level, human cognition is naturally capable of completing them as well.

We note the obvious difference that exists between people’s abilities to perform cognitive tasks relevant to the creative process, for example the difference between an expert and a layperson. We posit, however, that the difference in skill level between expert and layperson is much smaller than the difference between a layperson’s skill and the skill of state-of-the-art computational modules for many tasks that are useful to the creative process. If the difference between the state-of-the-art module and a layperson’s performance is small, then a human-delegated module is likely not worth developing for that task.

The first step in implementing ideal-module prototyping is to identify the system’s modules and their inputs and outputs. Care should be taken to specify each human-delegated input and output exactly so that that information is as similar as possible to what the computer module processes. Each human-delegated module must be designed to be displayed and answered via a human-friendly UI. Care should be taken with any instructions issued to the human delegate to ensure that the human has neither more nor less information than the computer module when completing the task.

It is also important to account for differences between how people and computers answer questions, especially for modules that compute tasks with no input that reflects knowledge of larger system goals or previously computed tasks. When interacting with such modules, people will naturally remember previous task inputs which could influence their responses. For example, the story-writing system we experimented with presents human delegates with analogy completion tasks, the outputs of which are used as input for later analogy tasks. If one person were to fill in these successive analogies in sequence, memory of previous tasks could influence their answers in a way that deviates from the computer module’s stateless operation. This effect can be mitigated by randomizing sequential tasks to reduce the likelihood that one person’s responses will be influenced by memory.

Various approaches can be taken to designing a user interface that facilitates human participation in the creative system. For example, in our experiment with applying IMP to a story-writing system, we built a client-server web interface for participants to interact with. This design allowed for widespread deployment via the Internet to attract a diverse set of participants to the experiment.

In order to design user interfaces that accomplish the researchers’ design goals, useful techniques may be drawn from the broader field of HCI. Lessons from Wizard-of-Oz techniques (Fiedler, Gabsdil, and Horacek 2004), which human-delegated modules may resemble from a UI or interaction perspective, and participatory design (Muller and...
Druin 2007), which concerns inter-domain experiences, may be of particular use. We reiterate, however, that the goal of applying IMP is to validate the design of an ultimately computational creative system; human-delegated modules will be replaced by computational modules in the final system.

Once all human-delegated modules have been created, they can then be connected with the same high-level information flow that the system would use if it had computer-controlled modules. In that way, the complete human-delegated system computes the same algorithm as a purely computer-controlled version of the same system.

In the case that the human-delegated modules are designed to match an existing creative system, it is likely that the high-level information flow will need to be re-engineered to accommodate for the asynchronous or event-driven nature of human-computer interaction.

Finally, as the human-delegated systems implemented under this paradigm are intended for evaluation and diagnostics, the systems should include logging of useful data that is generated as the program runs. In addition to the final system output, any other data that will give system designers insight into the system’s operation should be collected. Input and output data from each module are likely to be very useful for design analysis because they can be compared directly with corresponding data from computational versions of those modules.

Running the Human-delegated System

After the system is complete, the next step is to recruit people to participate in completing tasks for the system’s modules. This may be approached in whatever way the researchers believe will yield the best results for their system. Commonly, the modules in a computationally creative system compute tasks that are relatively simple for humans to complete. This should permit the recruitment of participants from a wide range of backgrounds and education levels.

We note that concerns such as population size and statistical significance are likely of lesser importance when using IMP to evaluate the performance of a creative system. At the design stage, such evaluation is largely subjective and depends on the researchers’ goals for their system. In this way, testing a system’s design with human-delegated modules is similar to running a pilot study.

The number of participants to recruit depends mainly on estimating how many people it will take to provide a sufficient quantity of diverse module outputs to complete system execution in a reasonable time period. It is a near certainty that the human-delegated system will take more time to run than a purely computational system. However, we anticipate that with ideal, human-delegated modules a small number of complete program runs should be sufficient to collect useful data with which to evaluate the creative system.

Once the participants have been recruited and the creative system has finished delegating tasks to them, the results can be analyzed. With proper data collection, the results should give insight into the performance of the system’s individual modules as well as its holistic performance.

The primary question is whether the creative system’s output is satisfactory. Although this criterion wholly depends on the goals that the researchers have for the system, the data collected by this method should aid any assessment of whether the system achieves those goals.

If the output is satisfactory, then the researchers can move forward with engineering a purely computational version of the system, armed with the knowledge that their algorithmic design is valid. Alternatively, if the output is unsatisfactory, the more granular data collected during execution can be used to inform design improvements.

Data should be collected for each module in the system, for example by recording all inputs and outputs to each module during execution. This will allow for analysis of individual module performance, as well as diagnosis of how information flows between modules. Recall that although similar analysis could be carried out on a purely computational system, IMP features ideal modules that eliminate the conflation of flaw-causing behavior between imperfect modules and flawed information flow.

If the individual modules’ inputs and outputs are found to be satisfactory, then unsatisfactory system output may be the result of flaws in the information flow between modules or require remediation via the modification, addition, or subtraction of modules. In the former case, it may be possible to redesign the connections between modules and simulate execution using the recorded input/output data, alleviating the need to recruit participants again. If that is not possible or if other modules are needed, partial reuse of the recorded data may still reduce the burden on participants.

It is possible that a given creative problem is not factorizable into modular tasks. Analysis of whether a given problem is factorizable or not is outside the scope of this work. However, we note that it may be useful to investigate the distinctions between problems that arise from an unfactorizable creative problem, the incorrect factorization of a factorizable problem, and software bugs in the implementations of a correctly factorized problem.

An alternative result of the evaluation is that one or more modules do not perform satisfactorily. If the modules’ tasks are delegated to humans, their performance should represent a cognitive ideal. Thus, any performance failings in the modules that cannot be attributed to UI flaws warrant careful consideration from the researchers. Useful questions to ask may include what the purpose of the module is, why calculation of that operation is out of reach for humans, and whether it is tractable to build a computer module to compute something that humans are apparently incapable of. It may also be useful to consider whether an ideal computational module exists to complete the task, in which case that should be used instead of a human-delegated one in both the prototype and the final system.

The nuanced outcomes of evaluating a creative system in this manner show the power of ideal-module prototyping for teasing apart complex, interconnected interactions between modules and accurately diagnosing the origins of low-quality output.

IMP Checklist

To summarize IMP, we present a checklist of steps that should be taken to apply the paradigm to a creative system,
whether it is being designed or is already implemented. This checklist should be applicable to any modular creative system, and we encourage researchers to tailor its application to the needs of their system.

1. Identify computational modules in the system.
2. Implement human-delegated versions of all modules for which no ideal computational module exists, and engineer a version of the system that uses those modules.
3. Experiment with the human-delegated system by recruiting participants and having them complete the delegated tasks. Log useful data generated during the experiment.

The result of applying this paradigm is a set of experimental data that can be used in various ways to validate the system’s design or identify shortcomings in the system’s modules or information flow. Based on the results of the experiment, it may be useful to apply IMP again after modifying the system’s design.

**HIEROS Experiment**

In this section, we give an example of how ideal-module prototyping can be applied to an existing creative system. It is instructive to consider such an example both as an exemplar of how to apply IMP and a concrete example of what can be learned from doing so.

**HIEROS**

HIEROS (Spendlove and Ventura 2020) is a computationally creative system that writes six-word stories, a genre of microfiction that exhibits several properties that make it interesting for study. Their short length reduces the amount of data that needs to be generated and allows for rapid human evaluation of results. Despite their brevity, however, they are far from a trivial creative domain; effective six-word stories push the limits of semantic and linguistic constructs to deliver impactful or emotional experiences to readers. Six-word stories are also not far removed from raw analogy and semantics. Because it does not contain filler or unnecessary words, the quality of a six-word story is closely related to the quality of the underlying relationships between the story’s words.

HIEROS takes advantage of that final property to inform its modular system design. Two of HIEROS’ three modules select the words that will form a six-word story, and the third assigns a score to the story that reflects its quality. The remainder of the HIEROS algorithm passes information between these same modules to search for higher-scoring stories. Consequently, the success of the system is very closely related to the quality of its modules, making it an ideal candidate for evaluating using our ideal-module paradigm.

Figure 1 shows a diagram of how information flows between HIEROS’ modules. We will describe this information flow and each module in turn.

Using human-written exemplar six-word stories scraped from the web, HIEROS infers an underlying format for each story. This format includes a hierarchy graph of the words in the story computed using dependency parse information provided by the Stanford Parser (De Marneffe, MacCartney, and Manning 2006). In this acyclic hierarchy graph, each word is connected by an edge to a parent word—as determined by the dependency parse—except for a single root word that has no parent.

The relationship between each parent and child word is used to guide the selection of a new child word when presented with a new parent word. This task can be thought of as a classic analogy completion question $A : B :: C : ?$, with the original parent and child taking the place of $A$ and $B$, respectively, and the new parent taking the place of $C$.

HIEROS uses an analogy completion module to accomplish this task. The inputs to the module are words $A$, $B$, and $C$. The module’s output is a word $D$ that fulfills the analogy $A : B :: C : D$ and shares the same part of speech as $B$. An optional input word $E$ can be specified as a word that should be excluded for consideration as output.

HIEROS generates new stories by first selecting a human-written exemplar whose hierarchy graph will provide the format for generating a new story. It then selects a new root word via another module. The root word selection module simply takes a part-of-speech indicator as input and returns
a word of the appropriate part of speech on which to base the story.

Starting with the newly selected root word, the analogy completion module computes analogies for each edge between the root word and its children in the selected format’s hierarchy graph. Then analogies are computed for those children’s children and so on until all six words of the new story have been selected. The words in this new six-word story are not the same as the words in the original, but the new words share similar relationships between one another as the words in the original do, as guided by the format’s hierarchy graph.

After generating a new story, HIEROS assigns it a score via a story scoring module. This module simply takes a story as input and returns a numerical score, for example a real number between 0 and 1. The module should assign high scores to stories of high quality and low scores to low-quality stories.

Once a story has been scored, it is placed into a priority queue, after which the highest-scoring story is dequeued and mutated to search for higher-scoring stories. A story is mutated by selecting a non-root node in its corresponding hierarchy graph, using its parent node to compute a new analogy for that word in the story via the analogy completion module, and recomputing analogies for all words lower in the hierarchy than the mutated word. The initial analogy recomputation provides the optional input word $E$ to the analogy module to specify that the output should not match the pre-mutation word.

The mutated story is scored by the story scoring module and inserted into the priority queue, after which the highest-scoring story is dequeued and mutated, and the process repeats.

By generating, scoring, and mutating stories in this manner, HIEROS refines its stories, searching for progressively higher scoring stories. If a specified number of mutation steps have passed without finding a higher scoring story, execution terminates, and the highest scoring story seen during execution is returned as the system’s creative output.

Thus, we completed step one of the IMP checklist. HIEROS’ three modules, as represented in Figure 1, are:

1. An analogy completion module which takes words $A$, $B$, $C$, and an optional word $E$, and returns a word $D$ such that $A : B :: C : D$ holds, $D \neq E$, and $D$ has the same part of speech as $B$.

2. A root word selection module which takes as input a part-of-speech indicator and returns a word of that part of speech that is an interesting word on which to base a story.

3. A story scoring module which takes a six-word story as input and returns a numerical score in a certain range, with high scores being assigned to high-quality stories.

Note that each of these modules computes a task that is difficult to rigorously define. What does it mean for the relation $A : B :: C : D$ to hold between four words? What is the definition of “an interesting word on which to base a story”? What criteria should be used to assign a high or low score to a story? The nature of these difficult-to-define questions makes these modules both scientifically interesting and difficult to compute. HIEROS’ computer modules that attempt to compute their respective tasks find some success but often fall short of ideal performance, and the system as a whole produces somewhat poor stories.

The key question, then, is whether low-quality output is the result of one or more flawed modules or is caused by a more systemic design issue. In order to more clearly diagnose HIEROS’ shortcomings, and to gather useful data that could be used to improve HIEROS’ design, we experimented with applying our ideal-module evaluation paradigm to the system.

Applying the Evaluation Paradigm

We re-engineered HIEROS to use human-delegated modules via a web browser UI and recruited participants to interact with it over the course of a week in order to test the system’s performance with ideal modules.

We created a simple HTML and Javascript web page that asks the participant to complete delegated tasks corresponding to HIEROS’ three modules: providing a root word, completing an analogy, or scoring a story. The web page connects to a server that runs the HIEROS algorithm and tells the web page which task to present to the user. Each task is used as module output in the system and advances the internal state as far as it can before requiring more tasks to be completed. After completing a task, the web page displays another task to the user, and users can complete as many as few tasks as they like.

Each task is worded to provide clear instructions to the user without providing extra information that could be incorporated into the response. The tasks are designed to be completed by any person who is fluent in English and can understand and follow the instructions. Tasks are not explicitly randomized, but the server does not immediately give users subsequent tasks that are generated from the previous task they completed. This prevents a user from completing the entire process of creating and scoring a single story.

The root selection task prompts the user to “enter an interesting word to base a story on”, including a desired part of speech.

The analogy completion task asks the user to “complete this analogy”, followed by the analogy presented in “$A:B::C:$” format, including specifying parts of speech. Neither the root word nor the analogy tasks enforce the part-of-speech specification; if the human user makes a mistake it is still considered valid input.

The scoring task presents a story and instructs the user to “score this story” with an integer score between 1 and 100, with no instructions pertaining to which aspects of the story to score. Integers were used instead of real numbers for ease-of-use.

Implementing these tasks as human-delegated modules constituted step two of the IMP checklist. Step three was to recruit participants and carry out the experiment.

To recruit participants, we distributed a link to the web page via social media and email. We did not collect demographic data. Users interacted with the web page over the course of a week, generating 142 stories total, including all
generated and mutated stories. To facilitate participant interaction, the human-delegated system did not terminate execution after a fixed number of mutation steps with no score increase, in contrast to the original HIEROS system.

Results

Examining the results of the human-delegated system, we find that the system’s stories are satisfactorily coherent, diverse, and interesting overall. This result provides strong evidence that HIEROS’ design is sound. Furthermore, comparing the results of the human-delegated system to the original computational system yields insights into which parts of the original system could be improved.

Although only the highest scoring story would be returned as its output, it is instructive to inspect all the stories the human-delegated system generated. The nature of the system’s scoring and mutation algorithm results in many stories that differ only by a few words. Filtering out very similar stories, the top four scoring stories from the human-delegated execution are:

Wilted. I Was a recent past.
Super chewy. Dry Food. Creature Meal.

The first story was created using the format of the web-scraped exemplar “Shit. I AM the adult supervision.” and demonstrates an interesting difference in subject matter while still adhering to similar inter-word relationships. The second and third stories also differ intelligently from their shared exemplar “Bitter taste. Swollen lips. Bye lover.” and are examples of two mutations of similar stories. The fourth story comes from the exemplar “Last human. Wrong Planet. Alien Delicacy.” and demonstrates how the food-related analogies present in the original influence the direction the new story takes after starting with the new root word “creature”.

Conversely, stories written by the original HIEROS system are less coherent and tend to represent unintelligent synonym substitutions for most words in the exemplar story. Examples follow, including the exemplar story in parenthesis for comparison:

Cried ourselves helplessly. Sobbed them asleep. (Cried myself asleep. Screamed myself awake.)
Ought Myself certainly rent one bullet? (Can I just buy one bullet?)
There gets unending voluptuousness from fecundity. (There is immense beauty in diversity.)
Be possibility to anybody that lives. (Be kind to everything that lives.)

Despite the low quality of the original system’s stories, the ideal system’s results demonstrate that HIEROS’ story writing algorithm could be a viable method for writing interesting and novel six-word stories. The only difference between the two systems is their modules, so it follows that the difference between their stories’ quality can be accounted for in the difference between the ideal modules and the computer modules.

The primary module that guides HIEROS’ creation of six-word stories is the analogy completer, which answers analogy questions of the form $A : B :: C : ?$. The computer module that HIEROS uses to solve these tasks is structured as follows.

A semantic vector is calculated for words $A$ and $B$ (the parent and child words from the hierarchy graph, respectively) by subtracting their word2vec word embeddings (Mikolov et al. 2013), effectively encoding the semantic relationship between the words. That semantic vector can then be added to the embedding of a new parent word $C$ to produce a new child word $D$ that shares the same semantic relationship, i.e. it satisfies the analogy $A : B :: C : D$. Word $D$ is then returned as the module’s output.

Examining the original system’s output, we observe that the computer analogy completer often simply returns synonyms of $B$ (the word in the original story to be replaced) as its output. The module does not seem to account for the relationship between $A$ and $B$ and how it could be applied to $C$, even though word2vec’s embeddings are often touted as facilitating analogical reasoning via simple geometric operations.

Meanwhile, the input and output data we collected from the human analogy completion module demonstrates a more nuanced and intelligent selection of analogous words. Three groups of example analogies demonstrate patterns in how participants completed this task.

The first group are analogies that cleverly apply an aspect of the relation between $A$ and $B$ to the word $C$ to derive an output word $D$. Examples of this group include:

- taste : bitter :: outcast : lonely
- lips : swollen :: wildernesses : overgrown
- smiles : someone :: rots : tree

A second group are analogies in which the left-hand side is not interesting or evocative. In this case, participants often fell back onto choosing a synonym of $B$, as displayed in these examples:

- tempted : twice :: expired : once
- luck : still :: tears : seldom

Another interesting group of responses seems to occur when there is a meaningful relationship between $A$ and $B$, but not one that can be easily applied to $C$. In this case, participants chose words that were primarily related to $C$:

- smiles : today :: eat : bread
- supervision : adult :: past : history
- went : anyway :: sought : fearlessly

All the data collected via this experimentation method is open to interpretation and is intended to aid the researcher in evaluating and improving their own system. As such, we do not make any strong, statistical claims about the results, we merely report our subjective observations of useful trends.

HIEROS’ scoring module is of critical importance to the system’s overall performance. Even if the system’s story
generation is somewhat flawed, an accurate scoring module could still guide the system to high-quality output.

HIEROS’ scoring module succeeds in assigning low scores to low-quality stories but is unable to consistently assign high scores to high-quality stories. This results in a large variance between scores assigned to generated stories, confusing the system’s ability to search the space of similar stories for higher quality stories.

The data collected from the human-delegated scoring module reveals that participants were not as consistent in their scoring as they were with analogy completion. Scoring stories on a numeric scale with no guidance or context is a difficult task for humans, and it was presented without modification in order to mirror the task that the computer module computed.

Quality differences between two stories are highly subjective, making it difficult to analyze small differences between human-assigned scores. Examination of broader trends in the human-assigned scores, however, seems to indicate that high-quality stories are indeed assigned higher scores than low-quality stories. Compared to the previously presented highest-scoring stories, the following lowest-scoring stories demonstrate a markedly lower quality level:

Swallow the biscuit tree rots eternity.
Wilted. History Was a zookeeper penguin.

Aside from the expected result that humans are accurate story scorers, a more nuanced insight emerges from examining patterns in the stories generated by the human-delegated system.

Because the HIEROS algorithm selects stories to mutate based on score, the system generated many stories that were mutations of the same story. When a new story was introduced, its first generation was often not of the same quality as the current high-scoring story which had been iterated and improved upon. This resulted in new stories never having a chance to be refined because the system ignored them in favor of the reigning champion.

This suggests that a division be made between scored stories as they compete to be mutated. Instead of a single priority queue organized by score, the system would likely be improved by keeping a separate list of newly generated stories from which stories are selected at random to mutate and score. This would give newly-generated stories a number of iterations to be refined before being discarded.

HIEROS could still maintain a priority queue but only use it for refined stories whose scores surpass a certain threshold. This more nuanced refinement algorithm should prevent the priority queue from being dominated by already-refined stories that exclude new stories and deny them the opportunity to be refined.

We note that this insight was only made possible by the ideal module experiment; HIEROS’ original scoring module was not skilled enough to make such fine distinctions. By running the system with ideal, human-delegated modules we were able to gather unique information that informs a better system design.

A New Prototyping Paradigm

Prototyping is an invaluable tool in any design discipline (Thomke and Nimgade 2000; Gerber 2010). It allows designers to evaluate a design while it is still in the early stages of development and avoid expending time and resources on implementing a design that will ultimately prove unsatisfactory.

Computational creativity is a challenging field that requires the design of complex systems. We argue that ideal-module prototyping represents a powerful method for evaluating designs of new and existing creative systems and that adoption of this paradigm will benefit the computational creativity community.

Improving Creative System Design

Ideal-module evaluation can aid the improvement of a wide variety of creative systems in different ways depending on the nature of their designs. Systems that attempt to improve the results of an already functioning creative system, complete systems with unsatisfactory results, and entirely novel systems can all benefit from applying this design evaluation paradigm.

Meta-analysis of computational creativity research, such as that conducted by Colton and Wiggins (2012), reveals recurring patterns in the development of creative systems. Colton and Wiggins describe one such observation that they call the latent heat effect, which describes the phenomenon that “as the creative responsibility given to systems increases, the value of its output does not (initially) increase”. As increasing the creative responsibility of computational systems is an implicit goal of research in this field, it behooves us to better understand and reason about this phenomenon.

Ideal-module prototyping provides a tool with which to probe and validate system designs, and applying it to designs that take on increased creative responsibility will allow researchers to better diagnose the causes of the latent heat effect. Increasing such responsibilities likely involves the addition of new modules to a system or a significant modification to the system’s information flow. Applying IMP to the newly designed system will allow the designers to verify that those changes did not invalidate the system’s capability to generate high-quality output when operating with ideal modules.

Thus, this new paradigm can help explain the latent heat effect and provide a strong argument that although a system with increased responsibility currently produces inferior output, it has the potential to generate high-quality output in the future.

We anticipate that IMP evaluation will be most beneficial when it is applied early in the design of a creative system. Embarking on the development of a new system requires trust that the design will prove fruitful in the end. Experimenting with an ideal-module prototype before investing development effort into building complex new computational modules allows researchers to determine ahead of time whether it will be worth expending that effort.

Building ideal, human-delegated modules requires precise definitions of those modules’ interfaces. This forces de-
signers to carefully define the exact responsibilities of those modules early in design, an exercise that is beneficial to system designers in its own right.

In addition to saving time by alerting researchers to unproductive design routes early, careful application of this paradigm could result in a system in which human-delegated modules can be swapped in place for computational versions without requiring changes to the higher-level control logic, thereby reducing development time.

Although IMP is primarily intended to be applied early in design, existing creative systems with lackluster output can benefit from its application as well. As detailed in our experiment with HIEROS, this method can provide more exact insight into why a system is failing and give its designers data that they can use to improve its design. We are hopeful that by accurately diagnosing their flaws and pointing the way toward how to improve them, the introduction of this paradigm will aid in rescuing unsatisfactory creative systems that have previously been shelved.

Collaboration & Modular Systems

Beyond application to single creative systems, we anticipate that adopting IMP will have a positive effect on the computational creativity research community as a whole.

The requisite thresholds of trust and confidence required to invest resources into developing a creative system are multiplied when dealing with collaborative research projects. IMP provides a means by which such confidence can be established before serious collaborative investment is made. Researchers seeking collaboration could first evaluate their design via this method to demonstrate its validity. Alternatively, this method opens up opportunities for useful replication or analysis of previously presented creative systems.

Modular designs lend tractability and flexibility to computationally creative systems. Researchers in the computational creativity community should design modular creative systems with precisely defined module interfaces and responsibilities. In addition to simply being better design practice, this will allow researchers to take advantage of the power of applying ideal-module prototyping to their designs.

Modular designs also present unique opportunities for collaboration. If multiple creative systems are designed to use an identical module, any effort spent developing and refining that module will pay off multiple times. Existing computational modules such as word2vec, WordNet (Miller 1995), and GPT-2 (Radford et al. 2019) are examples of modules that are widely used by different systems. By employing this ideal-module prototyping to validate shared-module systems before developing such computational modules, all involved parties can have assurance that their efforts will not be in vain.

Conclusion

The computational creativity community would benefit from a robust prototyping tool that allows creative system designs to be validated before putting in the challenging effort to implement them computationally. We have presented ideal-module prototyping (IMP), an evaluation paradigm for modular systems that compares imperfect computational modules with ideal human-delegated versions.

IMP can be applied to new system designs or retroactively to existing systems, as we demonstrated with our experiments with HIEROS. The results of the human-delegated system allow researchers to make strong claims about the validity of their system designs or accurately diagnose flaws in existing modular systems.

Researchers in the computational creativity community should design modular creative systems with well-defined module interfaces and employ ideal-module prototyping to validate their designs. Adopting IMP will allow for the improvement of existing creative systems, greater confidence in the development of new systems, and increased opportunities for collaboration.

References


Artificial Creative Intelligence: Breaking the Imitation Barrier

Rowland Chen, Roger B. Dannenberg, Bhiksha Raj, and Rita Singh
School of Computer Science and Language Technologies Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213 USA
rowlandc@andrew.cmu.edu
rbd@cs.cmu.edu
bhiksha@cs.cmu.edu
rsingh@cs.cmu.edu

Abstract
Not all knowledge is created equal. A hierarchical architecture is a method to classify knowledge for use in the field of human cognition and computational creativity. This paper introduces an Insight-Knowledge Object (IKO) model as a framework for Artificial Creative Intelligence (ACI), a step forward in the pursuit of replicating general human intelligence with computing machinery. The IKO model represents a novel approach to computational creativity and computing machine design.

Introduction
Since the dawn of computing machinery (Jacquard’s first use of wooden punch cards for looms to imitate and weave fabric designs in 1801, from Zimmerman 2017), the challenge of achieving true machine creativity remains elusive (IBM 2019; Olenik 2019). Creativity has been defined as an idea that is novel, surprising, and valuable (Boden 2004; Abraham 2018; Runcon 2012; Stein 1953). Recent advances in artificial intelligence, machine learning in particular, approach the successful imitation of the styles of select painters through style transfer in the visual arts (Zhou, Z. et al., 2019). Advances have also been achieved in voice impersonation (Andabi 2017; Gao 2018). However, the authors assert that true creations do not emerge from style transfer. Nothing truly novel and no surprises result from imitation. Similar to Turing’s question from 1949, “Can machines think?” (Turing 1950) the authors approach their ultimate question, “Can machines create?”

A fundamental re-thinking of human and artificial creative processes and corresponding mathematics and computing machinery are indicated to break through an imitation barrier. This paper introduces an Insight-Knowledge Object (IKO) model as a framework for human creativity and Artificial Creative Intelligence (ACI), a step forward in the pursuit of replicating general human intelligence with computing machinery. The IKO model represents a novel approach to computational creativity and computing machine design.

Insight-Knowledge Object Model
A proposed Artificial Creative Intelligence framework, the Insight-Knowledge Object (IKO) model (Figure 1), builds on Chen’s prior work (Chen 2009). Proposed here is a way of modeling the human thought process that can be used to shape the development of ACI machines. It is important that each level in the IKO hierarchy be present in the process of human thinking as well as in artificial attempts at reproducing that process. Knowledge objects are on the lefthand side, and insight processes appear on the right. In the IKO model, as insight processes and knowledge objects ladder up through the hierarchy, increasing levels of cognitive sophistication are reached. The IKO model has eleven levels of knowledge objects and ten levels of insight processes acting upon the knowledge objects. The IKO model begins with a state of knowledge object known as void. In this state, even the acknowledgement of nothing does not exist. At the very top of the IKO model is the knowledge object creation generated by an inspirational insight process.

Descriptions of Each Insight Process
Instinctual insight acts on the void to generate null, the
first true layer of knowledge. Null emerges from void as primal instincts create an awareness of one’s environment. This level of knowledge is deemed null because at this level, there is at least consciousness of existence or non-existence. In the void, even consciousness does not exist.

Definitional insight acts on the null to elevate knowledge into data. At this level, the insight process gives definition to objects and actions in the null. Definitional insight labels a collection of unnamed and unidentified things so that distinctions are drawn between them. Each object is now defined and becomes a datum.

Contextual insight acts on data to generate facts in the next layer of the hierarchy. Facts represent a richer and fuller set of knowledge than pure data. For example, if one takes the word coffee as a datum there is no context for reference. Given some context such as the commodities trading market, coffee takes on the meaning of a traded good. If food service is the context, then coffee takes on the meaning of a beverage. Contextual insight allows distinctions to be made between data to create different facts.

Utilitarian insight acts on facts to generate know-how, how an object is used and for what purpose. In our coffee example, utilitarian insight emerges to provide the know-how for what to do with coffee. In the commodities market context, know-how would be how to trade coffee on the spot or futures markets. In the food service context, know-how would be how to prepare coffee for consumption. Without utilitarian insight, coffee has no real value. Simply speaking, utilitarian insight provides knowledge of use.

Experiential insight acts on know-how to generate memories. The execution of know-how generates experiences that can be remembered and used in the future. Following the coffee example, experience in making coffee enables a barista to remember how much foam to put on top of a latte.

Reflective insight acts on memories to generate wisdom. Reflection works on a meta-plane of thinking and takes on a new layer of abstraction in the knowledge hierarchy. Insights are not simply generated on single points of execution but a set of memories. For example, remembering how to make a latte is a memory but digging deep to understand why people order lattes takes reflection. Wisdom emerges as a person can take a step back to reflect and learn from prior thoughts, decisions, and actions.

Recognitional insights act on wisdom to generate patterns. This insight function is in the realm of data science and data analytics. Recognizing patterns links related or unrelated pieces of wisdom to generate knowledge that would not emerge otherwise. For example, connecting the preparation of a perfect latte to the film “Seven Samurai” (Kurosawa 1954) in which one of the samurai has dedicated his whole life to perfect his skills as a swordsman represents connecting two topics that on the surface are completely unrelated. A pattern emerges with the recognition that these humans continually strive for mastery in their respective fields. Recognitional insights create connections that drive thinking further and produce patterns.

Extrapolative insights act on patterns to generate predictions. This insight function is in the realm of statistics and probability. Making predictions links related or unrelated patterns to generate knowledge that would not emerge otherwise. Extrapolative insights produce predictions used for weather forecasting, social media user preference, and the serving up of relevant advertising. Again, in our coffee example, extrapolative insight is required to predict the demand for lattes in a coffee shop during the course of a day.

Comparative insights act on predictions to generate imitations. This insight function is in the realm of machine learning and implementations such as generative adversarial networks (GANs). Predictions are compared against a reference with the goal of achieving an imitation that most closely matches the reference. Much work and many examples of painter style matching, voice impersonation, and literature already exist. This level of the knowledge hierarchy is the present boundary of today’s state-of-the-art deep learning techniques. One could surmise researchers and scientists have hit an imitation barrier. For lattes, skilled
baristas can copy fanciful milk designs on the surface of lattes based on prior examples of a master barista’s work. This is latte style transfer.

Inspirational insights act on imitations to generate creations. This insight function has not yet been designed and requires fundamental research across the disciplines of neuroscience, physiology, psychology, and computer science. The proposed ACI framework surpasses the imitation level by using unique, to-be-developed computing machines and computational algorithms, simulating human inspirational insights, to output creations. A master barista thinks up and executes unique and novel designs for lattes (Figure 2). She is not simply mimicking prior art. The inspiration process of the proposed IKO model breaks through the imitation barrier. At this level of the IKO hierarchy, the challenge of Artificial Creative Intelligence could be solved.

Figure 2: Inspired latte design (Photo credit: 123rf.com)

Related Work – the DIKW Pyramid

In information management and business technology, knowledge management was a new approach for data storage and data processing that was introduced late last century. A hierarchical model was developed at the time to capture, store and retrieve documents (origin unknown, Sharma, 2008). Data, information, knowledge, and wisdom comprise the model, also known as the DIKW pyramid (Zeleny, 2005; Rowley, 2006). In the knowledge management field of information technology, the DIKW pyramid has not evolved since its introduction (Elliott 1934). A collateral benefit of the new IKO model, primarily intended for artificial creative intelligence applications, is a potential new methodology for knowledge management in business information management and information technology.

The IKO Model as Functions

The increasing levels of sophistication in the IKO model can be written as a set of functions in which insight processes act upon knowledge objects (variables) of the form:

\[ KO_i = f(KO_{i-1}) \]

The increasing levels of sophistication in the IKO model are symbolically presented below where a function is an insight process. The exact nature of the functions has not yet been discovered. The use of the “\( f(KO) \)” notation is intended to drive the thinking of those seeking ways to represent creativity and perhaps, to provide a way to develop the mathematics of creativity.

\[
\begin{align*}
\text{void} & = \text{initial condition} \\
\text{null} & = f_1 (\text{void}) \\
\text{datum} & = f_2 (\text{null}) \\
\text{fact} & = f_3 (\text{datum}) \\
\text{know-how} & = f_4 (\text{fact}) \\
\text{memory} & = f_5 (\text{know-how}) \\
\text{wisdom} & = f_6 (\text{memory}) \\
\text{pattern} & = f_7 (\text{wisdom}) \\
\text{prediction} & = f_8 (\text{pattern}) \\
\text{imitation} & = f_9 (\text{prediction}) \\
\text{creation} & = f_{10} (\text{imitation})
\end{align*}
\]

Therefore, creation is a function of imitation, prediction, pattern, wisdom, memory, know-how, fact, datum, null, and void.

The authors postulate the system function, \( F \), relies on ten nested, or composite, functions. If any function is skipped, the system collapses, and output falls short of a creation. \( F \) can thus be written as:

\[
F = f_{10} \circ f_9 \circ f_8 \circ f_7 \circ f_6 \circ f_5 \circ f_4 \circ f_3 \circ f_2 \circ f_1
\]

Breaking the Imitation Barrier

“Deep learning and current AI, if you are really honest, has a lot of limitations.”

Jerome Pesenti, Vice President of Artificial Intelligence, Facebook (December 4, 2019)

“Creativity and where the authors started exploring with the (Disney movie trailer) is fascinating because deep learning isn’t the answer to creativity.”

John Smith, IBM Fellow at IBM Research (2019)

Current attempts at computational imitation rely on machine learning techniques that use algorithms acting on training and testing data in a resource-intensive approach. In the context of the proposed IKO model, generative adversarial networks (GANs) produce imitations (level ten) acting directly on data (level three), thus skipping several levels. But imitations are not creations. The authors hypothesize the existence of an imitation barrier that limits current approaches to level ten of the model (Figure 3). The authors also hypothesize that as a result of level skipping in the context of the IKO framework, attempts at artificial creativity...
will not be able to break through an imitation barrier to reach creations.

![Image of a house of cards collapsing](Photo credit: 123rf.com)

**Figure 3: The Imitation Barrier**

Computational creativity occurs in natural brain function, and the proposed IKO hierarchy suggests a model for human thought processes. When a human creates, inspiration does not happen in a vacuum. All knowledge in the brain comes to play. For example, facts, know-how, memories, and wisdom are needed but are left out of current imitative computing. Refer to the system function, F, that is a composite of all insight functions. If any function is skipped, the system collapses.

![Image of a house of cards collapsing](Photo credit: 123rf.com)

**Figure 4: House of cards collapsing**

JerryBot: the IKO Model in Practice

In various music genres, creative improvisation on a single or multiple instruments delivers music that is pleasing to the listener. The music genres include rock, jazz, jazz fusion, classical, blues, rhythm & blues, bluegrass, Indian, and others. The improvisation technique of one member, or more members, of a musical group creates new, real-time compositions during a live performance or in a music studio. Music improvisation is an example of human creativity – novel, surprising, and valuable (Boden 2004; Abraham 2018; Runcon 2012; Stein 1953). The sounds created are so valuable (pleasing) that a music artist develops a fan base comprised of music lovers around the world. The improvised music is created through inspirational insights based on the intelligence, experience, and technique of the music artist or artists.

When a highly skilled and highly loved music artist passes away, his or her new improvisational music creations and corresponding fan experiences disappear forever. An Artificial Creative Intelligence machine fills the emptiness left behind by a dead artist by recreating the improvisation style and reproducing the instrument sounds of the specific artist. The music fan’s pleasurable listening experience with the late artist and his or her band is extended. The goal of the machine is to perform alongside human musicians in real-time, and this hybrid human-artificial band’s music is indistinguishable from any new music that could have been performed if the artist were not dead. One can call this the creation game, with apologies to Turing (Turing, 1950). A version of an artificial music improvisation machine design appears in Figure 5.

In this system, called JerryBot (Chen 2020), multiple living musicians improvise with the ACI machine, similar to a completely human band. On-going work at Carnegie Mellon University focuses on a specific musician, the late guitarist Jerry Garcia of the rock group The Grateful Dead. A database containing recordings of all live concert performances of the Grateful Dead has been captured from the Internet Archive (Internet Archive, 2019). These data, which the authors call DeadNET (Figure 6), represent nearly 2,500 concerts, performed between 1965 and 1995, and approximately 7,500 hours of music. The data comprise the Historical Variations of Song. For example, within DeadNET, there are over six hundred variations of the song “Playing in the Band.”

In the block diagram, one can see recognition, extrapolative, comparative, and inspirational insight functions highlighted in yellow. These four insight functions are integrated into a single system in an attempt at developing an ACI machine. Additionally, embedded in the recordings captured in DeadNET are the band’s musical data, facts, know-how, memories, and wisdom accumulated over 30 years of per-
formance. With DeadNET and the insight functions of recognition, extrapolation, comparison, and inspiration, the ACI machine under development includes all levels of knowledge object and insight function. No levels are skipped. The machine holistically brings together a whole-brain approach to computational creativity.

The authors have adopted music as a language paradigm to leverage existing solutions and tools developed in the Language Technologies Institute at Carnegie Mellon University as well as open-source software. In process are the use of a “bag of words” model and vector quantization of .wav files (Figure 7, Raj et al. 2019). Continuing work is intended to address the other components of the artificial music improvisation machine, for example, the isolation of Jerry Garcia’s lead guitar track from the recordings of the entire band to create JerryNET perhaps modeled after the neural basis of auditory attention used for speaker isolation (Geravanchizadeh 2020), discovering the neural basis of inspirational insight, modeling the music conversations that occur when a jam band assembles and plays, and formulating how a machine can self-assess the value of computationally-created music (pleasantness of the output from a hybrid human-machine band). Much work remains.

**Other Use Cases**

Beyond music, other potential applications for Artificial Creative Intelligence include use cases in language arts (the haiku problem) and autonomous vehicles (control and decision-making in unlearned situations).
The Haiku Problem

5-7-5. That is the syllable count for the Japanese three-line poem style known as haiku. In haiku, the first two lines set the tone of the poem and the emotions of the reader. The third line is a surprising closure. For example, below is a haiku by Edo-period (18th century A.D.) Japanese artist Katsushika Hokusai:

*I write, erase, write,
Erase, re-write again, then
A red poppy blooms*

The final line of the haiku is a surprise and an emotionally satisfying ending created by Hokusai in a moment of inspiration. The third line also diverts the reader from extrapolation and a prediction of what might conclude the poem.

The haiku problem is for a machine to generate a third line that is both surprising and pleasing to the reader. Attempts at machine-written haiku have been able to achieve the correct syllable count (for example, in Aguiar 2019). However, further work is required to achieve an elusive novel close.

The No-Win Self-Driving Car Problem

A continuous flow of conscious choices exists in driving situations. Even in relatively stable conditions, for example, cruising down the highway, humans choose speed, acceleration/deceleration, and direction. Add in entertainment, and the choices grow. Currently, autonomous vehicles integrate robotics, image recognition, comparative insight, and deep learning. However, a vehicle’s response to turbulence introduced into the environment is limited to learned scenarios. But what about a situation unanticipated and unlearned by the autonomous systems?

An example of this would be a scenario in which multiple externalities are introduced into the moving vehicle environment to create a no-win situation. How would a self-driving car, with a passenger, make a decision between crashing into an 80-year old man, a pregnant woman, or a toddler or driving off a cliff (there being no other alternatives)? The controller of the vehicle has several choices to make, be the controller a human or an AI system. As Nyholm and Smids (Nyholm 2016) have written, this problem is being addressed in the development of accident algorithms, and it is not analogous to the trolley problem that comes from the study of ethics. The authors of this paper contend that currently, only a human can make an acceptable conscious choice in this no-win scenario, where acceptability might differ from culture to culture. The authors are also cognizant of the risk of being caught in a prisoner’s dilemma in pursuit of an artificial solution. An Artificial Creative Intelligence machine, developed for music improvisation, could provide the level of computational creativity required for making real-time choices to minimize physical, emotional, ethical, and societal damages.

Conclusion

Artificial Creative Intelligence requires a fundamental rethinking of how knowledge is managed and how intelligence is processed in vivo. The Insight-Knowledge Object model is offered as a new framework for scientists and engineers working in the fields of computer science, neuroscience, psychology, etc. Each level of knowledge object builds on the level beneath it. In order to continue up the hierarchy, knowledge must be captured and modeled differently from current and prior attempts at artificial creativity. In the IKO model, creativity is dependent on all levels of knowledge that come before it: void, null, datum, facts, know-how, memories, wisdom, patterns, predictions, and imitations. If any of these necessary levels are skipped, then the creative process collapses much like a house of cards, and Artificial Creative Intelligence remains out of reach.

Still unknown are the mathematics and implementations required to execute Artificial Creative Intelligence based on the proposed IKO architecture. The authors recognize blurry lines exist between levels that might direct future work in the direction of analog computing. To be determined is the exact nature of the insight functions. Are matrix algebra, the calculus, statistics, and probability sufficient to push the current AI boundary into creative insight processing? If not, what mathematics is needed for a software solution? How will quantum computing impact AI? Will an analog network of solid-state neurons play a role? Additionally, the biological basis for the IKO model has yet to be explored to understand the natural science foundation of the insight functions.

The issue of speed also requires solving. For ACI to be truly effective in numerous use cases, response times must match or improve upon that of human processing.

To be addressed as well is training an Artificial Creative Intelligence machine if deep neural networks are incorporated. Unsupervised training is particularly challenging in artificially judging pleasantness of newly generated music. Baker et al. (Baker, 2020) have launched an initiative in Human-Assisted Training-AI as a direction for future work in machine learning.

The above questions and issues require continued research and exploration across multiple disciplines including work in neuroscience, psychology, mathematics, computer science, electrical engineering, and philosophy.

The challenge is how to break the imitation barrier if at all possible.

Acknowledgements

In-kind support for this work was provided by Audio Intelligenz. The authors give thanks to the independent, anonymous reviewers of this paper for their feedback and suggestions.
References


Elliot, T.S. 1934. The rock. London: Faber & Faber


Where is the Life the authors have lost in living?
Where is the wisdom the authors have lost in knowledge?
Where is the knowledge the authors have lost in information?
T.S. Eliot, 1934. The Rock
Post-creativity and AI: Reverse-engineering our Conceptual Landscapes of Creativity

Jan Løhmann Stephensen
Department of Culture & Communication
Aarhus University
8000 Aarhus C, Denmark
aekjls@cc.au.dk

Abstract
This paper introduces the notion ‘post-creativity’ as a reference point for discussing the ways in which the computational simulation of creativity perpetually seems to hinge upon conceptions of creativity that are both much-too-human and non-contingent. Taking issue with the often implicit idea within the artificial creativity-agenda that creativity somehow exists before the fact, this paper, drawing upon Michel Foucault’s notion of the ‘dispositif’, insists that we must keep a steady eye on the historicity of the ideas and practices of ‘creativity’ in order to fully comprehend the ways in which computational/artificial creativity is part and parcel of the perpetual re-creation of creativity and hence, at best, contributing to a conceptual reverse-engineering of ‘creativity’.

Post-creativity
This paper takes its que from the notion ‘post-creativity’, an abbreviation of post-*anthropocentric* creativity, which hints at a notion of understanding of creativity that is not solely focusing on the human aspects of creativity.

An initial disclaimer is in order. This notion is not advocated in order to suggest that we have moved beyond creativity. It is not after creativity. The ‘post’-prefix rather hints at the human-centric tendencies within creativity studies and related disciplines and practices, including Computational Creativity (CC).

This becomes relevant as we are moving into an area in which the human factors are increasingly less at the centre of things, also in relation to creativity, and have been so for quite some time. This shift is the case both in relation to productive creative practices, where there are a lot of practical experiments happening these days. And it is the case in relation to our conceptual understandings of those practices we label ‘creative’. So, ‘post-creativity’ both has to do with changes in the making of stuff; and with the making sense of making stuff (and as I will make clear a bit later, these two levels — of making, and of making sense of making — are dialectically interwoven).

The main reason we need a notion like this now is the increasing entanglements of humans and machines in the making of art, especially various forms of art generated by or in collaboration with AI. Given that most art-making today at some level involves digital technologies, this actually happens much more often that we perhaps realize. Digital technologies tend to precondition our workflows and influence the aesthetic characteristics of the outcome of our creative efforts (e.g. Bardzell 2007). Or to be more precise: they precondition, influence and shape the efforts we engage in, whilst we understand and often explicitly frame them as being ‘creative’ (or as emerging out of ‘creative processes’).

Despite the proliferation of human-technology entanglements on numerous levels of so-called ‘everyday creativity’ (cf. Runco and Bahleda 1987; Richards 2010); and despite the fact that this kind of entanglement is quite interesting since these kinds of objects, as Bruno Latour has noted, “no matter how important, efficient, central, or necessary they may be, tend to recede into the background very fast [...] — and the greater their importance, the faster they disappear” (2005: 79-80); this paper will mostly focus on forms of creative art-making that originates in some variant of autonomously working AI — and does so explicitly. This paper will in other words engage with practices and assumptions about creativity that are on par with CC’s ambition to explore “the potential for computers to be autonomous creators in their own right” as it is stated in the call for papers for this year’s conference.1 Or as d’Inverno and McCormack have labelled it: ‘Heroic AI’, in

1 [http://computationalcreativity.net/iccc20/full-papers](http://computationalcreativity.net/iccc20/full-papers). It is, however, worth noting that this ambition is quite different from the one given in the "Welcome to the Eleventh International Conference…”1, which defines "computational systems” as entities that "exhibit behaviours that unbiased observers would deem to be creative." [http://computationalcreativity.net/iccc20](http://computationalcreativity.net/iccc20) Whereas one relates to being creative, the other relates to leaving the impression of being creative (although the next sentence gradually slides back into essentialist thinking: ”formalising what it means for software to be creative” (ibid.; my emphasis).
which “the software takes on the role of the lone creators” (2015: 2438).

I will focus on this agenda of autonomous artificial creativity whilst simultaneously being aware that at closer scrutiny, this perhaps never really happens at all. Setting up Generative Adversarial Networks (GAN) and appropriate training datasets or in other ways designing relevant “learning environments” for Deep/Machine Learning algorithms and so on inevitably means getting another host of humans involved (Yanisky-Ravid 2017). Though we do, of course, tend to think of these people as ‘non-creatives’, thus keeping up the impression (or ideal) of autonomous machinic creation; but that’s another story.

Yet, even if this constant re-entanglement with the human-social was a factor that could be dealt with — if we could somehow subtract all human engagement from creative computational processes all down the production line — I would still argue that a much more fundamental problem is at stake, namely the assumed existence of creativity as something already being “out there”, “in here”, “up there”, etc. In short: the implied existence (being) of an unartificial creativity, that either could somehow be uncovered in advance and then reproduced or artificially simulated; or which computational approaches could bring us closer to understanding — just to paraphrase the two most commons agendas for working with artificial creativity.

**Being Creative, Being Man**

Thus, the issue of creativity being (this or that) is closely related to one of the main reasons people often mention for finding art generated by AI particularly interesting, namely that it challenges our conception of what creativity is, and subsequently often also how creative products come into being, through which processes, etc. (Stephensen 2019). It allows us, or perhaps even forces us, it is often argued, to rethink the nature of human creativity and to pose fundamental questions such as “what does it mean to create?” and “how are things brought into being?”

But on top of that, it even forces us in much more general terms to rethink what it means to be human. This might, of course, come across as a tall claim, an overstatement. But if we look at the ways in which we over the recent half century (since World War 2 and in particular since the 1960s) have invented creativity it becomes obvious, that in our minds creativity has become one of those things that define who we are. Yet, in the long history of ideas, we have never before thought of ourselves in this way, as the creative being. For quite some time this was, in fact, an idea that quite easily would have gotten you in serious trouble with the Church and other authorities; perhaps most famously formulated by the Church Father St Augustine of Hippo (and subsequently repeated by St Thomas Aquinas) in his dictum *creatura non potest creare* (the creature, or the created being, cannot itself create) (Pope 2005: 45). Creativity was not the role of Man.

Now it is. Now, we think about ourselves in these terms: Man as the creative being. For numerous reasons, many of which have been socio-political and economic, creativity has suddenly become one of our essential defining qualities, a ‘species-characteristic’ or something at the core of our ‘species-being’ as young Marx (1967) would put it (cf. Fromm 1961; Wartenberg 1982; Sayers 2011). In short: creativity has become one of those things that seems to make us different from all other living beings; what makes up the essence of being human. And hence also, something we should strive for, individually and socially.

Once again it is important to emphasize that in a historical perspective this is something completely new. Michel Foucault (2003) once suggested that some ideas have become so natural to us that we forget they have a history; that at some time they were perhaps even unthinkable. The notion of creativity as a shared human faculty and as something essentially human is one of those ideas.

**Politicizing creativity**

Two historical trajectories seem important for grasping this conceptual shift. One is related to political issues. It is important to note that the burgeoning use of ‘creativity’ as a noun post-WW2 coincided with New Left, countercultural critiques of capitalism and its so-called alienating effects upon individuals as well as sociality at large. The critique of capitalism — much of which took its cue from the at the time recent release of young Marx’ hitherto relatively unknown writings on alienation in translation (collected in Fromm 1961) — had two intertwined effects:

1. To advocate the ideal of so-called creative productive activity against alienated labour and its organisation under capitalism, which was accused of stifling the former (often presented as the dichotomy ‘creativity’ versus ‘organisation’, ‘bureaucracy’, ‘capitalism’, etc.). And (2) to successfully, on a broader societal level, install the idea of creativity as not only a common feature of humanity at large. It is also one of those things that was crucial to individual self-actualization, that is: of de-alienating one’s own humanity. This was a perspective that was in line with Abraham Maslow and Carl Rogers who explicitly linked the actualization of one’s creative potentials to “becoming a person” (Reckwitz 2017: 149-152), which gradually has become common sense (in the Foucauldian sense). Now, “[we] are all creative, at least in potential!”, as Glavaneau and Kaufman without any hesitation state in the first line of their introductory chapter “Creativity” from the latest edition of *The Cambridge Handbook of Creativity* (2019: 9).

In recent years, in contrast to the critical use of the term that used to dominate in which creativity was pitted against labour under capitalism, these two tenets of creativity have mostly been talked about as something we might achieve — and increasingly expect/are expected to achieve — in or
at work (Morgan and Nelligan 2018).\footnote{The fetishism of creativity did not just spring from the counter-cultural critique of capitalism. Quite ironically it was accelerated by an energizing loop around the very powerhouse of capitalist consumer society, the advertising industry, which successfully managed to harvest consumer dissatisfaction to the service of more capitalist consumerism. ‘Creativity’ thus became central to what Thomas Frank (1997) has labelled ‘hip, anti-capitalist consumerism’, which advocated the idea of consumption as a cultural practice that was simultaneously creative and anti-capitalist.} Whereas creativity used to be linked to capitalism’s (economic, political, social) \textit{Other}, it has now become pivotal to an perpetually changing, all-encompassing economic system to which there is no real alternative (Fisher 2009). For instance, on the back-cover blurb on management guru Gary Hamel’s book \textit{The Future of Management} (2007), one of the most influential of the creativity-at-work advocates Richard Florida celebratorily sums up this shift as follows:

“For the past century, people have worked in the management prisons of Industrial Age — which has wasted the energy, creativity, and human potential of our people. Gary Hamel […] creates an inspiring and needed vision for the future of management that is not only more human, but can unleash the full potential in all of us.”

But it is not just business lingo. Throughout the second half of the 20th Century, all these ideas about the inherent commnunality, the emancipatory potential and not least the human实质性 of creativity were also at the centre of a host of avantgardistic artistic practices. With reference to the same cluster of political and philosophical ideas these movements sought in various ways to democratize or distribute creativity; either by inviting non-artists to participate in creative practices (cf. art historian John Roberts’ notion of the ‘opening of the circuits of authorship’ (2007)); by imploding the category of the artwork (cf. the readymade or process art); or within theory in more general terms: by applying the label “creative” to practices hitherto not thought of as such (cf. the tendency within Cultural Studies to reinterpret consumption as a creative practice (McGuigan 2011)).

\textbf{Democratizing, distributing, recombining creativity}

The other trajectory worth mentioning is related to technological innovations, more specifically the proliferation of new digital media technologies within the last two decades. Here, the aforementioned previous agendas of democratizing and distributing creativity have been (im)materially embedded into the functional architecture of “new media” (cf. Turner 2006, 2013; Manovich 2013).

Parallel to this, the practice of recombining already existing materials (\textit{creatio ex materia} — or as new media lingo labels it: remixing — was also consolidated as a genuine feat of creativity, which it had not been previously under the auspice of the romantically inclined idea of \textit{creatio ex nihilo} (Mason 2003; d’Inverno and McCormack 2015; Lessig 2008; Boden 2004).

In a similar vein, the proliferation of technologies of (creative) co-operation (Rheingold et al. 2005) have given prominence to a conception of creativity that emphasises co-production, collaboration, co-creativity, symbiotic relationships between producers and users, etc. (Jenkins 2006; Meikle & Young 2012; Bolter 2019). These architectures of co-operation and collaboration stand out as realizations of Tim Berners-Lee’s ambition for the World Wide Web as a non-hierarchical site of ‘intercreativity’ (1999: 169-172), only on a much grander scale than he could ever have imagined, of course, especially with Web 2.0 (or 3.0, 4.0 or where ever we are presently at).

This conception of creativity as a collaborative or distributed practice has also been adopted within artificial/computational creativity research. d’Inverno and McCormack have for instance suggested that so-called ‘Collaborative AI’ where “the system supports, challenges and provokes the creative activity of humans” seems the better option in comparison with ‘Heroic AI’ in which “the software takes on the role of the lone creators” (2015: 2438). Likewise, Davis et al. have suggested that thinking in lines of ‘computer colleagues’ as ‘co-creative agents [which] collaborate with humans in continuous real time improvisation to enlrich the creative process” seems most promising (2015: 110).

Despite the relevance and timeliness of downplaying the often overindulgent claims of the AI-hype, especially that concerning the prospects of a ‘Heroic AI’, both d’Inverno & McCormack and Davis et al. nonetheless end up aligning a bit too closely with the above-mentioned two trajectories, the socio-economic/political/ideological and the technological, and their combined legacy: the dominant conception of creativity as a profoundly human activity that, at least in principle, can be done by anyone, anywhere, in any given material, and often in concert. And on top of that with the idea being pitched to us by big tech firms like Apple (cf. their famous ‘Think Different’ campaign) or Adobe (‘Creative Clouds for Teams’), namely: that creativity could best be achieved by using, and not least purchasing, digital media technologies and software ecologies from these specific companies.

\textbf{AI and automation — creativity as our last refuge?}

In combination, and in accord with our new technological imaginaries, these two trajectories have further popularized the impression that creative self-realization, especially in one’s working life, is a fair and normal thing to expect of oneself, as well as of others (e.g. colleagues and employees). Which is a historically unprecedented expectation, which we have only quite recently come to embrace on an individual and societal level. Never before have we had
such high cultural expectations to the mundane activities of maintaining our subsistence.

If we for a moment return to the idea of Man as the creative being it seems fair to say that in recent years this idea has been getting ever more traction in relation to current socio-economic and political discussions on automation, robotisation and AI. Whilst these technologies might be able to replace us or even out-perform us when it comes to all the tedious stuff (in the so-called ‘realm of necessity’, as the older Marx labelled it in Capital (2010: 593)), the good news is, the argument often goes, that this will only leave us with more time in the ‘realm of freedom’ to do what we do best and which we are the only ones really capable of doing, namely: to be creative. This argument has for instance been expressed by Tobias Queisser, founder of the AI film management system Cinelytic which recently signed a deal with Warner Bros. in order to guide their decision-making at the greenlight stage through the application of Big Data, thus substantially influencing what gets to go into production:

“Artificial intelligence sounds scary. But right now, an AI cannot make any creative decisions. [...] What it is good at is crunching numbers and breaking down huge data sets and showing patterns that would not be visible to humans. But for creative decision-making, you still need experience and gut instinct.” (Siegel, 2020)

So, even in Queisser’s quite un-romantic take on it, creativity is still our last refuge, the ‘final frontier’ which no machine can conquer (cf. Colton and Wiggins 2012). But, once more, this is no new idea. Theodore Roszak for instance already discusses the prospect of computationally simulated creativity (so-called ‘objectified creativity’) in his The Making of a Counter Culture. Here, he notes that, “The most ominous aspect of such statements is the ever-present ‘yet’ that appears in them” (1969: 282) — cf. Queisser’s “right now” and “still” — after which Roszak quotes Rand Corporation-affiliated philosopher of technology Emmanuel G. Mesthene for having suggested that

“No technology as yet promises to duplicate human creativity, especially in the artistic sense, if only because we do not yet understand the conditions and functioning of creativity. (This is not to deny that computers can be useful aids to creative activity.)” (ibid.).

Regardless of whether we today think of this problem as genuinely new or not — I would, for instance, personally suggest that any talk about the “AI Winter” should be accompanied by a footnote on the ‘hibernation of same old arguments’ — it does seem that the prospect of the future development of a genuine artificial creativity that could someday even outperform us in this field as well, inherently raises huge philosophical questions (cf. Boden 2016: 119) about the nature of Man and his/her role in the grand scheme of things; not to mention enormous political problems as well.

(Re-)inventing and Simulating Creativity

As already suggested, one of the crucial, yet often much too implicit, assumptions underlying the urge and ambition to artificially simulate human creativity is the idea that human creativity exists as something that can be reproduced, simulated or emulated. But despite the ways in which we often tend to speak of creativity — namely; as something existing prior to us, with few, if any historical contingencies, as something objectively “out there”, “in here”, or “up there”, as something we can learn more about, define, nurture, increase or enhance (cf. Sternberg 2019), perhaps even replicate or simulate, etc. — it is a creature of our own invention.

So, what do I mean by saying we have invented creativity? Well, of course, I am not saying that we did not invent or create stuff earlier. We certainly did. But we did not frame and ascribe it the same kind of importance or value. We did not praise it, strive for it, or encourage it like we do today; and we did not organize our individual lives, our collective practices or our societies in relation to it, build institutions for it, or have journals or conferences to discuss it either.

Readers familiar with Michel Foucault (1977) would have noticed that this kind of thinking is in line with his notion of the ‘dispositif’ (or ‘apparatus’). This argument has also been made by German sociologist Andreas Reckwitz in The Invention of Creativity (2017), where he specifically develops the notion of the ‘creativity dispositif’ which, as a specifically Modern constellation of practices, modes of knowledge and sensibilities, pivots around the production, appreciation and institutionalization of perpetual novelty, invention and innovation.

This is, in fact, also one of the important points in Yuval Noah Harari’s bestselling book Sapiens: A Brief History of Humankind. In the long history of humanity, we have only quite recently come to value, let alone systematise, inventiveness and the ability to make something new; which — dialectically, you might say — has greatly accelerated the pace of invention itself, especially in the West (2014: 314-315). Or as Neil Postman famously quipped: “the greatest invention […] was the idea of invention itself” (1993: 42).
But we have not just invented creativity in the singular — even though that is how we usually refer to it. We have, in fact, invented many creativities⁴ in numerous forms and guises: Big-C creativity and little-c creativity (Gardner 1993); P-creativity and H-creativity as well as combinational; exploratory and transformational creativity (all from Boden 2004); process, person, product and so-called pressure-oriented creativity (Rhodes 1961); creatio ex nihilo and creatio ex materia (Mason 2003); etc. And on top of that, we have typically described or defined these multiple creativities either in so vague, generic terms that they seem almost meaningless, which allows us to project all kinds of idiosyncratic imaginaries onto them (Hentig 1998); or in so reductive terms that they all leave out a lot of artefacts, phenomena or practices, which most of us would have thought should have been included.

Hence, the philosopher of art Morris Weitz (1956) would characterize “creativity” as an ‘open concept’. But perhaps the term ‘essentially contested concept’ — with strong emphasis on ‘contest’ — would be even more appropriate (Stephensen 2019)? Given the ways in which the idea, concept and quite diverse practices of ‘creativity’ over the last decades have become entangled in innovation policy-making, human resource management, economics, urban planning, education system reforms, etc., it does seem safe to say, that those days of creativity’s (seemingly) ideological innocence are long gone; if, of course, they ever where here at all.

Reverse-engineering Creativity

All of this does, of course, when we seek to create, build or program an artificial creativity, immediately spur further questions such as: “Exactly which one of the many creativities are we building?”, “What are we specifically simulating?”, or “What non-artificiality are we making an artificial version of?” And perhaps even more critical: “According to which agenda?”, “What kind of problems are we trying to solve?”, “Whose problems?”, “To whose benefit?”, etc.

The problem is this, I would argue, that we are trying to build something we do not know what is. But not because we have not yet figured out what this mysterious, wondrous thing called creativity really is, which, of course, is the illusio (Bourdieu 1980) that secures the involvement of all the different stakeholders within the emerging booming academic field of creativity studies (including the various fields of artificial/computational creativity). No, it is rather because we constantly tend to forget the contingency of ‘creativity’ itself — including the fact that it is still under constant re-construction.

This in turn means that the result of simulating human creativity computationally, algorithmically or artificially might simply end up in a re-modelling of our conceptual notion(s) of creativity based on what we can do computationally. Much like what has happened to our definitions of ‘intelligence’, which, as Edwin G. Boring (1923) once famously noted, is what we can measure with intelligence tests. Applied to AI-building this could easily surmise to something like intelligence is simply what we can simulate (cf. Smith 2019). Within the domain of artificial creativity this line of thinking would then end up something like creativity is what we can simulate (or compute).

Could this really be? Well, in fact, this kind of reverse re-conceptualization has already happened before. This is how our various ‘creativities’ have been invented and reinvented throughout the 20th Century. Take for instance the currently prevailing definitions, which are all mostly variations on the idea of creativity as the production of useful or relevant, novel or original ideas, products or services (Amabile 1997; Howkins 2001; Florida 2002), which all seem defined in reverse from business interests, in particular those based on Intellectual Property Rights in which locating a single point of origin (or at least: of sufficient original variation) is pivotal to the process of ascribing ownership (Lessig 2004; McKenna 2011). Similarly, before these business and innovation-centric definitions came to dominate, we witnessed the emergence of the idea of creativity as a species-characteristic (cf. the New Left, countercultural readings of young Marx’ economic manuscripts), which sprang from an ideologically charged contestation of the elitist idea of creativity as the genius’ prerogative, often in tandem with the critique of the alienating effects of labour under capitalism. So, claiming that the future successful creation of an artificial creativity will merely be a (conceptual) reorientation based on what is computationally possible is perhaps not all that unreasonable. This kind of conceptual reverse engineering (in lack of better words) has happened before. It will happen again.

So, my argument goes, this is the same logic that is at stake when computational creativity researchers for instance point to improvisation and pattern-prediction as being closely related, seek to understand creativity through “heuristic search, analogical and meta-level reasoning” (once again quoted from ICCC’s call for paper), or define creativity as the interplay between style, which can be imitated/simulated, and constraints, which can be applied at will (Pachet 2018).

I should, however, emphasize that I am not saying this in a normative sense. I am not shouting from the rooftops that “you [or we] are ruining creativity!” or something to that
effect. I am not driven by some romantic-essentialist urge to defend Creativity with a capital-C nor a more genuine and/or politically potent version of it. Of course, this argument has recently been voiced in various differences by for instance Mould (2018), Pasquinelli (2019) and Fazi (2019), who all assume the existence of a more genuine creativity being suppressed by a false one. And once again, exactly this position could also to be found already in Roszak, who on the very idea that human creativity can be objectified computationally noted that

“[t]he presumption involved in such statements is almost comic. For the man who thinks that creativity might yet become a technology is the man who stands no chance of ever understanding what creativity is. But we can be sure the technicians will eventually find us a bad mechanized substitute and persuade themselves that it is the real thing.” (Roszak 1969: 282)

My point is another. We simply need to be aware what it is that we have achieved when, or if, it finally happens: we will most likely have accommodated our notions of creativity to fit what is technologically attainable. We will have performed a conceptual reverse engineering. Or to be more precise: we will have given more conceptual credit to those “practices of creativity” that fit what we can do technologically. And less credit to other practices, of course.

The reason why this is important is that such a conceptual shift might change our ways of recognizing, appreciating and rewarding something as being creative in general, especially given the current hype around anything remotely AI-related. And since creativity as a set of practices or outcomes on the one hand, and creativity as a norm or a set of values on the other, are so closely interrelated, this will in turn instigate new sets of practices that we might term ‘creative’ — and make other practices less likely — also beyond the “ghettos” of AI-related creativity conducted by experimental scientists. In short: we will also on the productive, practical level have reinvented invention, re-created creativity. Which may, or may not, be problematic on various levels. But we of all people, as scientists and researchers, should acknowledge that this is what has occurred, that this is what we have achieved. And certainly, we should not confuse it with something quite different.

Re-inscribing Man

Perhaps even more paradoxically, in that very process we will also have revived crucial tenets of an older notion of creativity, namely the one heralding the heroic, often male, autonomous subject (Proudfoot 2010). There simply seems to be an inherent anthropomorphic strain within our thinking about machines, not least in relation to AI and how we interpret and try to make sense of what these technologies can do — and a lot of so-called “wishful mnemonics” going around as well. But perhaps Joanna Zylinska has put it with most clarity in a recent talk on her upcoming book AI Art: Machine Vision and Warped Dreams:

“this frequently posted question ‘can robots or can computers be creative?’ […] reveals itself to be rather reductive because it’s premised on a pre-technological idea of the human as a standalone subject of decision and action.” (Zylinska 2019: 18:33-18:50)

Rather, Zylinska argues, much in accord with my own notion of ‘post-creativity’ mentioned at the beginning of this paper, humans are always already intimately entangled with and heavily conditioned by nonhuman materialities which are also part and parcel of those processes we label ‘creative’. There is, in other words, no stand-alone human creativity.

There is, of course, quite an ironic twist to this. On the one hand, the project of artificial creativity is in accordance with recently emerging theoretical trends — even within the strictly anthropocentric parts of creativity studies (cf. Fox and Alldred 2017: 77-95). Thus, it often emphasizes both the collaborative (social) and re-combinatory (remix) character of digital media-afforded creative practices; often also by subscribing to theoretical vocabularies that emphasize the intimate entanglements of humans/nonhuman entities in creative practices etc. On the other hand, much discourse on and practice in relation to AI and creativity immediately seems to revoke this distribution of creative agency to nonhuman entities. Hence, creativity becomes all-too-human once again, that is: an artificial, computational emulation of a supposedly already existing non-artificial, purely human capacity or practice.

Why research computational creativity?

So, why study computational or artificial creativity at all, then? Well, my take on it would be to hold it at greater arm’s-length. Even though it does not tell us anything about what creativity is per se — or how it works in a raw state, so to speak — studying the endeavours to achieve it will tell us other things. It might tell us what creativity means to us. It will tell us what we (individually and collectively, even on a societal scale) expect and hope from it, in which settings and situations we do so, and so on. This is important because what we code, program, realize or enact when we build systems of artificial or computational creativity is, in fact, the creativity dispositive rather than

---

5 Cf. the ways in which much work on autonomous artificial creativity seeks to emulate the Masters, often including a fairly traditional, organic aesthetic language or style, which has led Joanna Zylinska to refer to much AI-generated art as “imitation art” based on the “pointless production of difference” (2019: 37:34)

6 Proudfoot borrows the notion ‘wishful mnemonics’ from McDermott (1970), who criticized how computer scientists discredited their own reputation by ascribing human characteristics (feelings, intensions etc.) to machines and codes. ‘Creativity’ would, of course, rank high among these.
creativity as such. In building and designing these systems, we not only install certain conceptions of what we regard as creativity (and by implication: what we do not regard as creative); we also install — albeit often implicitly — certain rationales for doing this (why we find this or that creativity important).

This does, for instance, in turn beg us to ask a host of other questions: “Why do we continuously seem so hell-bent on reproducing ourselves in our material environment?” Or for that sake: “Why should creativity be made easier, more accessible or even automated?” And perhaps even more fundamental questions pop up — which is also relevant to those who claim that we should learn to appreciate the creativity of algorithms or the nonhuman in more general terms (cf. Gervas 2010; Colton and Wiggins 2012): “Why are we even under the impression that we need more creativity? Are we really short of it? Is novelty, inventions or innovation (all these ‘products’ of creativity) what we lack on a societal or individual level?” In short, studying the outcomes, the processes and the discourses on computational creativity really offers us an opportunity to study the creativity dispositif as it works its way through us, including how we have become almost addicted to the very idea of creativity.

So, finally, what does the notion of ‘post-creativity’ contribute to our study of this particular field? In my opinion, it allows us to think beyond purely anthropocentric understandings of creativity, not only under present conditions (in relation to creativity-enhancing software and AI), but even more profoundly, in a historical perspective, thus enabling us with the possibility of critically grasping how ‘creativity’ has always already been both contingent and materially embedded, and continues to be so. Computational creativity is merely the latest instalment in this story.

References

Boring, E.G. 1923. Intelligence as the Tests Test It. New Republic 36: 35-37


Explainable Computational Creativity

Maria Teresa Llano
SensiLab, Monash University
Melbourne, Australia

Mark d’Inverno
Goldsmiths, University of London
London, United Kingdom

Matthew Yee-King
Goldsmiths, University of London
London, United Kingdom

Jon McCormack
SensiLab, Monash University
Melbourne, Australia

Alon Ilsar
SensiLab, Monash University
Melbourne, Australia

Alison Pease
Dundee University
Dundee, United Kingdom

Simon Colton
SensiLab, Monash University
Melbourne, Australia

Abstract

Human collaboration with systems within the Computational Creativity (CC) field is often restricted to shallow interactions, where the creative processes, of systems and humans alike, are carried out in isolation, without any (or little) intervention from the user, and without any discussion about how the unfolding decisions are taking place. Fruitful co-creation requires a sustained ongoing interaction that can include discussions of ideas, comparisons to previous/other works, incremental improvements and revisions, etc. For these interactions, communication is an intrinsic factor. This means giving a voice to CC systems and enabling two-way communication channels between them and their users so that they can: explain their processes and decisions, support their ideas so that these are given serious consideration by their creative collaborators, and learn from these discussions to further improve their creative processes. For this, we propose a set of design principles for CC systems that aim at supporting greater co-creation and collaboration with their human collaborators.

Introduction

Although systems from the field of Computational Creativity (which we will refer to from now on as CC systems) – and more generally AI systems – have been successful in different application domains, a common challenge for users and researchers is that most of these systems behave as black boxes, limited to opaque interactions where processes and reasoning are completely unknown or obscure (Scherer 2016). As a result, users are left questioning the nature of the system’s decisions, or are discouraged to the notion of collaborating with them. These limitations have raised the need for the development of models that offer more clarity and transparency in order to improve the potential for human-machine interactions with CC systems (Muggleton et al. 2018; Bryson and Winfield 2017).

The field of Explainable AI (XAI) has grown in recent years with the goal of making black box systems more transparent and accountable through models of explanation that communicate the way decisions have been reached. Current approaches to XAI, which are commonly associated with popular but opaque machine learning methods, centre on providing explanations as part of the output of the system; i.e. the focus is on delivering a final result to a user alongside a rationale of how this result was created. However, the creative process is often performed in isolation, with no place for intermediate explanations as the process progresses, let alone place for exchanges of information that can exploit human-machine co-creation.

In this paper we propose Explainable Computational Creativity (XCC) as a subfield of XAI. The focus of this subfield is the study of bidirectional explainable models in the context of computational creativity – where the term explainable is used with a broader sense to cover not only one shot-style explanations, but also for co-creative interventions that involve dialogue-style communications. More precisely, XCC investigates the design of CC systems that can communicate and explain their processes, decisions and ideas throughout the creative process in ways that are comprehensible to both humans and machines. The ultimate goal of XCC is to open up two-way communication channels between humans and CC systems in order to foster co-creation and improve the quality, depth and usefulness of collaborations between them.

Designing and implementing this type of communication is extremely complex; however, we believe it is important to start a discussion towards what is needed for more fruitful and productive partnerships between CC systems and their users. Creativity is not a solo act, it is a social activity that benefits from life experiences, from influences of people, and from the contributions of different collaborators. This is why we are interested here in co-creative CC systems and on addressing the lack of two-way channels of communication. This may result in increased novelty, or higher-value creative collaborations, however it is difficult to anticipate how successful and in which way any creative collaboration can emerge, but we believe the kind of active collaborations proposed through XCC would ultimately enhance the human-machine collaboration experience and increase the engagement of users with CC systems.

In the rest of the paper, we first provide relevant background literature. Then we present an overview of the current state of the art of co-creation and explainable AI in general and in CC in particular. We follow by identifying design principles that we believe CC systems with explainable capabilities should have. We finish with a discussion outlining challenges that need to be considered, opportunities that may arise, conclusions and directions for future work.
Motivation and Related Work

Co-creation, real-time interactions and collaboration are important topics in the CC community; however, the communication between CC systems and their users is limited to a few exchanges, or the rationale behind their individual actions is often unknown by either, or there is little or no opportunity for discussion of the ideas presented, and many system outputs are discarded without a second thought.

Take for instance the SpeakeSystem: a real-time interactive music improviser which takes as its input an audio stream from a monophonic instrument, and produces as its output a sequence of musical note events which can be used to control a synthesizer. The BBC Radio 3 Jazz Line Up programme commissioned SpeakeSystem in 2015 for a live human and computer performance with alto saxophone player Martin Speake\textsuperscript{12}. Figure 1 shows a recording of the performance that was uploaded to a timeline annotation system in order to carry out a focused discussion about it. The conversation involved the musician, the algorithm designer and a member of the audience. However, a key participant of the performance was not involved, the interactive music improviser. To illustrate, at some point of the recording the human musician commented:

“I think by this stage that I wanted the algorithm to come up with a new interaction mode.”

while an audience member said:

“I wonder what you are both thinking going into this section. The algorithm not a lot I suspect! Otherwise it would play notes.”

Even though the algorithm designer provides information about how the system operates, he is unable to provide an explanation of what was going on at specific points in the performance as only the system “knows” the details of what went on at every point of the performance.

If the SpeakeSystem had been equipped with communication capabilities, not only could it have provided insights about its creative process in the conversation above, but the human musician could have communicated his intentions during the performance (perhaps through some kind of visual or haptic signal), giving the system a chance to respond during co-creation. This could have resulted in a more engaging experience, for the musician and the audience alike.

Co-creation, Real-Time Interactions and Collaboration in CC

Models to improve collaboration and co-creation within CC have been explored in the community. In (Davis et al. 2015), for instance, an enactive model of creativity is proposed for collaboration and co-creation. The key principle for this model is that the system has a level of awareness and that co-creation happens through a real-time and improvised interaction with the environment and other agents. Another approach is mixed-initiative co-creativity (Yannakakis, Liapis, and Alexopoulos 2014), which exploits a bi-directional communication based on the collective exploration of the design space and human lateral decisions that are used by the system to guide the creative task. Although these models establish communication channels, these are limited to an action/reaction type of model; i.e. there is little opportunity for further introspection in addition that they do not enable the systems to further support their contributions.

The You Can’t Know my Mind installation of The Painting Fool (Colton and Ventura 2014) presented real-time interactions with users that resembled some of the aspects we cover in this paper: the system provides explanations about its process, motivations, etc. (e.g. by providing a commentary alongside its output), the user provides content to guide the creative process (e.g. by expressing a particular emotion to be the inspiration for the painting), and the system utilizes visual cues to reveal what is going on (e.g. with the use of an on-screen hand while it paints the picture). Although these interactions simulate communication, it is mostly a one-way creative endeavor.

The Beyond The Fence musical is also an important sample of human-machine collaboration in the field. The musical writers that were involved in this effort were enthusiastic about the possibilities of collaboration with CC systems; however, they highlighted some of the flaws they faced when working with them:
We waded through probably a thousand pages of computer generated tunes to find the fragments and phrases that felt right for the show’s needs” Musical writers comment in (Colton et al. 2016)

“Collaborating with computers is utterly unlike anything either of us have encountered before, and at times, it has been incredibly frustrating” Musical writers comment in (Colton et al. 2016)

These comments highlight that ultimately, for a long-standing (working) human-machine relationship to be sustainable in the context of creativity, there is a need for mechanisms that enable a more active partnership. As the musical writers put it:

“I rather think that the future holds ways of allowing human artists to work with computers more comfortably, and with more control of their output, ultimately to support and perhaps shape their own creativity in ways they might not have been able to envisage” Musical writers comment in (Colton et al. 2016)

The need for communication

Nickerson et al. (Nickerson, Elkind, and Carbonell 1968) described the increasing complexity of human-computer interactions based on the ability to communicate with one another “the thing that, above all others, makes the man-computer interaction different from the interaction that occurs in other man-machine systems is the fact that the former has the nature of a dialogue”. This thought has been echoed by other researchers, who have emphasised that computers are ‘comparable’ to humans in some dimensions when seen as collaborators, particularly as dialogue-partners (Kammersgaard 1988).

With the rapid growth of AI techniques, this discussion has gradually highlighted the existence of a higher level of intelligence when contrasting the views of ‘interaction as tool use’ and ‘interaction as dialogue’ (Hornbæk and Oulasvirta 2017), arguing that utility and usefulness are the main aspects for the first type of interaction to work, while having a constant, simple, direct and natural communication and understanding between human and computer, is key for the second type of interaction to work.

This distinction of interacting with a machine considered as a creative intelligence rather than a tool is a key motivation for this work. For this, we need to consider a broader set of aspects of human interaction that are otherwise ignored in the narrower view of systems as tools. For this broader view, exposing the creative process is crucial. Specifically we hypothesise that establishing two-way communication channels, within a co-creative human-machine partnership, where the creative process is transparent as well as discussed, would improve interactions, build up trust on CC systems and encourage human engagement.

Overview on the state of the art of XAI in CC

In (Zhu et al. 2018) the authors defined a new area of XAI which they called Explainable AI for Designers (XAID). The objective of XAID is to support games designers in specific design tasks. In their work they identified three spectra that describe co-creation in the setting of games design: i) spectrum of explainability, which ranges from understanding of of the underlying operation of AI techniques to understanding of the input-output pattern, ii) spectrum of initiative, which refers to the level of intervention of the system (ranging from a passive tool to an active collaborator), and iii) spectrum of domain overlap, which is concerned with the degree of co-creativity that is needed (defined in terms of overlap of shared tasks).

Relevant to this discussion is also the concept of framing as proposed in (Cook et al. 2019). Framing, as has been applied to date within the CC community, is mostly intended as a ‘final interaction’; i.e. to accompany an output with the expectation that it will increase its perceptive value. However, in (Cook et al. 2019) the authors propose advocacy and argumentation as a potential purpose for framing.

Both of the works mentioned above are relevant to our work and are a step towards the vision of CC systems playing a more active role in creative collaborations; however, XCC focus spreads not only to co-creation but also to other interactions that occur when producing a creative act; such as setting up an initial goal, delivering the product to a final user, producing feedback, etc. Moreover, we ground the interventions of a CC system not only on the current act of creation in which it is involved, but also in past experiences (i.e. we propose that CC systems should have a memory of their work). Additionally, we also adopt the notion of argumentation and advocacy as a role for XCC; but our model, proposes this role not only as a way to support a creative artefact, but also as a way to increase their involvement since the conception of an idea towards the production of it.

In the next section we outline some design principles for systems with XCC capabilities. We use a running example to illustrate our ideas using linguistic communication as the primary medium for explainability; however, we consider communication in its broader sense, not just through linguistic forms. We will expand on this later in the paper.

Design Principles for XCC Systems

The main objective of the design principles outlined next is to enable both CC systems and their human users to communicate with each other so that there is a common and clear understanding throughout their interactions. We have drawn from a range of research that looks at human collaboration, teamwork, cognitive science and psychology, as well as from our experience on the development of computational creative systems. From this, we have identified four main design principles: mental models, long-term memory, argumentation and exposing the creative process. We now explain each principle in detail.

Mental models: Are representations of key elements of the creative environment that help conceptualize, understand and construct expectations of how things work and how individuals interact within a creative collaboration (McCormack et al. 2020; Mohammed, Ferzandi, and Hamilton 2010). The concept of a mental model comes from psychology and cognitive science research (Craik 1967; Johnston-Laird 1993) and has more recently been successful
in HCI (Norman 1982; Krug 2014). Having a good mental model of how software works is vital for its usability.

**Relevance to CC:** The idea of mental models as a key aspect for the design of real-time co-creative systems has been highlighted previously in (McCormack et al. 2020). Mental models consider a broad set of aspects of human interactions that would aid the understanding of essential elements within a collaboration and of the interactions throughout it. CC systems can use these representations, not only to understand the operation of their co-creators, the environment and the domain, but also to reason about them and seek different (creative) ways of interaction by questioning them; for instance by playing a bossa nova style chord progression when improvising with a musician with a strong rock background (i.e. challenging preferences). Shared mental models can increase CC systems’ awareness of processes within the collaboration. For instance, in order to achieve cognitive convergence when the mental models of the collaborators do not match (e.g. when they are not moving towards the same goal) (Fuller and Magerko 2010).

Equipping CC systems with mental models, of both themselves and of their co-creators, not only would enable better coordination when trying to come up with something new, but would also provide a valuable resource for CC systems to explain, justify and defend their contributions. We believe this increases the capacity, of both CC systems and their users to generate appropriate and complementary output. As pointed out in (Brühlmann et al. 2018), people are more motivated to use a specific technology when it is congruent with their personal values, goals, and needs.

**Features:** Relevant elements of mental models would involve team aspects such as goals, roles, capabilities, expectations, etc., domain aspects such as conventions governing the operation of particular domains, stakeholders profiles, relationships between elements of the domain, etc., as well as interpersonal aspects of individuals such as preferences and how they communicate. Sharing the underlying representation of a mental model would depend highly on the domain at hand. The dynamics of collaborations in different domains influence the way mental models are built and used. For instance, improvisational settings are governed by implicit interactions with subtle signs and cues, while in an advertising campaign setting, for instance, participants can explicitly communicate their intentions, discuss their ideas and establish agreements.

**Example:** Imagine for instance the following interaction between an Advertising Executive (AE) and an XCC system (XCCS) when collaborating to design an advert for a toothpaste (goal) in which the clients would like to emphasise the ideas of teeth and decay (domain information):

**XCCS:** How about the image in Fig 2? Teeth and dice are similar, as they are white, cube-shaped and shiny. Dice connects to gambling, which connects to poker, so I got the idea: “What if someone gambled with their teeth instead of with money?”

**AE:** I don’t like it, because it’s too big of a jump – the connection to toothpaste is not obvious.

The system can pick up from the explicit intervention that the AE does not like ideas whose connections are not obvious and can adjust its mental model with this information in order to generate new ideas.

**Long-term memory:** Is the capacity of storing and accessing information of past experiences and interactions.

**Relevance to CC:** Studies in cognitive psychology have found that memory is a crucial element of creativity and that an important part of the creative process happens by drawing on previous experiences and the information we have in our memories (Madore, Addis, and Schacter 2015). Being able to explain their decisions and support its ideas requires CC systems to have a memory of the processes, decisions and interactions they have undertaken in current and previous collaborations. “The act of remembering is an attempt to recreate events and experiences that have occurred in the past” (Stein 1989). Equipping CC systems with this ability would enhance the creative capacity of their collaborations. Possessing a memory would also serve other purposes such as breaking habits and avoiding repetition or mistakes.

**Features:** Enabling a bi-directional communication would provide opportunities for CC systems to store different types of information, such as failed attempts, successful artefacts, strategies used, temporal information, users’ reactions, etc. How to store and access this information is an important aspect to consider here. For instance, one methodology could be as defined in (Davis et al. 2015), where the authors use the concept of perceptual logic to classify information in a way that aids co-creation as follows: in local perceptual logic the system only considers specific details (such as a line in a drawing), in the regional perceptual logic the information is grouped into clusters (e.g. straight lines, lines that are close to each other, etc.), while in the global perceptual logic, the system considers relationships between regions (e.g. identifying that the left hand side of a drawing has fewer lines than the right hand side). Depending on the domain and the purpose of a CC system in that domain, different mechanisms may use to handle such memory.

**Example:** Let us take for instance the example of the AE and the XCCS working on the toothpaste campaign. The AE did not like the idea for the ad even after the XCCS provided an explanation. The system can then review past experiences and find additional strategies to support its idea, in doing so the system finds that in a previous interaction another AE used a tag line to clarify an abstract concept for an ad. The interaction then follows:

**XCCS:** I think the connection is good. There is a strong
surface similarity between teeth and dice and the idea is surprising as the two concepts are not normally associated; maybe we could add a tagline to support the image? Something related to the not apparent connection between teeth, gambling and money?

AE: That is good. I like the idea of using a tagline. Maybe something like “There are some things you just can’t afford to gamble with”.

AE: There may be something here, but I’m not sure yet...

Argumentation: Is the process of reasoning-about and supporting specific contributions within a creative collaboration.

Relevance to CC: A fruitful creative partnership enables participants to both explain and justify their ideas in order to unveil or clarify their creative value. An artefact may be of poor quality but the process behind it novel and interesting. CC systems that cannot further support and champion their creative contributions carry the risk of being prematurely discarded.

Features: As pointed out in (Cook et al. 2019), argumentation provides a set of very valuable resources that can be used by CC systems to enhance communication with their users. Take for instance the theory of critical questions (Walton, Reed, and Macagno 2008), which helps anticipate questions or concerns that may arise in specific situations with different stakeholders, as well as the mechanisms to try and address possible conflicts that these questions or concerns may arise within a creative collaboration.

Ultimately this part of the process involves an attribution or cognitive process that requires different dimensions to be taken into account. Is it a one-shot argument, or is it the start of a dialogue? Is the argument targeted for a co-creator, an end user, a member of the audience? What elements of the mental model should most influence the selection of arguments? When is the right moment to push or to stop? What kind of language, visualisation or medium should be used to convey the argument? What does the system want to achieve with it: clarification, persuasion, feedback, etc.?

Example: Take for instance the example of the AE and the XCC system in the toothpaste advert campaign. After the second interaction, when the system proposes the use of a tagline to clarify the image, the AE is willing to give a chance to the system’s idea. A further interaction could go this way:

XCCS: The idea and image are somewhat repulsive and repulsive adverts have a shock value which help people remember them. Other repulsive adverts have been quite effective. Have a look at this: https://www.youtube.com/watch?v=AOph5V78oxs (video showing an advert warning about the addictive nature of cocaine).

AE: Hmmmm... possibly.

Without the system pushing back a little, it is easy to imagine that a user may stop engaging after the first generative act, thus missing out on the concept.

Exposing the creative process: consists of opening up the environment and exposing the steps, assessments, metrics, influences, etc. that constitute the processes and decisions within the operation of a domain.

Relevance to CC: Providing an explanation for a process or a decision is a useful way to obtain a better understanding of what is going on within a closed environment; however, descriptions or clues envisaged for this purpose may sometimes fall short in aiding that understanding. Opening up the environment and exposing the steps, assessments, influences, etc. that constitute those processes and decisions would provide a clearer view of how a CC system works.

We believe that observing, feeling, or in some way sensing the underlying structures of a CC system, instead of being told how things work, may trigger thought processes in the mind of their co-creators that may result in benefits for the creative collaboration.

Features: Exposing the creative process requires the models and mechanisms used by a system to be interpretable so that they can be easily and unambiguously communicated to others. The development of interpretable models is being encouraged in the AI community because of the inherent problems with unexplainable models, such as unfaithful accounts of their computations (Rudin 2019). In the absence of interpretable models, CC systems should be equipped with interfaces that make communication with their users interpretable. In other words, the way a CC system communicates must be simple and precise, yet the communication needs to be meaningful so that the new information helps progress the collaboration. An example of this is provided in (McCormack et al. 2019) where the authors equip an AI musician (whose underlying model consists of a neural network) with the ability to continuously communicate how confident it feels during an improvised performance. Human performers also implicitly communicated their confidence to the computer via biometric signals. The work showed that this type of simple, interpretable, communication increased the flow within the human-AI collaboration and the quality of the music produced.

Example: Let us take for instance the example of the AE and the XCC system working on the toothpaste campaign. The AE is still not sure about the idea for the ad. The system can then expose its reasoning even further so that the AE has a better understanding where the ideas come from. Imagine for instance the CC system has an interface that allows the user to investigate the underlying structures behind its ideas through the mean of a visual graph representation of the system’s knowledge base.

By seeing the connections among the concepts in the knowledge base, the user realises that the two related concepts; i.e. dice and teeth, are not actually directly connected and sees this as an opportunity to improve the concept of the ad. The AE proceeds as follows:

AE: I still don’t like the image. How about blending the two similar concepts: dice and teeth?

XCCS: Ok. I’ve made the images in Fig 3. I prefer the first one, as the cube shape is clearer.

AE: I prefer that one too. I’ll adapt it and add tagline, brand and product information... Here is the final advert, in Fig 4.
A framework for XCC

Figure 5 summarises the kind of interactions of the proposed approach based on the four principles presented in this paper. Instantiating these interactions naturally depends on the creative domain, the stakeholders involved, and the stage of the creative process where the interventions occur.

The domain is the most important factor to determine the kind of medium for communication. In music, for instance, communication may most likely occur through the music itself – imagine for instance a CC musical composer demonstrating through a virtual keyboard how a pianist should emphasise a particular phrase –, while in painting this communication may occur through brush strokes – imagine now a CC painter-collaborator that paints all over a section of a painting they think should be emphasised (McCormack et al. 2020).

The stakeholders greatly influence the sources and type of information that these interactions would require. A co-creator may need technical details of the operation of a system – an ideation system may provide a tree representation of the relevant concepts from the knowledge base from which an idea was produced (as in the advertising example), while for an end-user the intuition would probably be more useful – for instance a CC poetry composer that explains the mood reflected in the poem because it read sad news in the newspaper (the end-user does not need to know, for instance, the technical mechanism of sentiment analysis used).

The stage of the creative process is most influential on how the elements of the design principles are managed. In a preparatory stage, when the collaboration is just starting, the interactions between CC systems and their users help set up the context of what the collaboration is about; i.e. a shared mental model is agreed – take for instance the motivating example of the SpeakeSystem: this stage would have allowed the system to inform the human musician about its reset function, which could have avoided the human musician wondering why the system didn’t make sense at some points during the performance. In a co-creation stage the interactions require an iterative process that involves a constant revision of the mental model in order to ensure that the collaboration is converging towards the same goal, the addition of new memories (which reflect the experience of the current interactions), access to old memories, and possibly various cycles of generative acts – here for instance, the human musician working with the SpeakeSystem could have signalled during the performance (through a visual cue or facial expression) when he wanted the system to challenge him instead of only responding to him. Finally, a post-creative stage provides an outlet to present the artefact as well as for feedback and reflexion. Such an outlet may also represent an opportunity for the revision of the mental models and for adding new memories to the system – here for instance the SpeakeSystem could have provided an explanation to the member of the audience who wanted to know what was going on at a certain point of the performance:

**Human annotator:** “I wonder what you are both thinking going into this section. The algorithm not a lot I suspect! Otherwise it would play notes”

**System:** I was enjoying what my partner was playing here. He/she seemed to really like this piece. He was very confident, was “in the groove”

As previously mentioned, we consider communication in its broadest sense, not just through linguistic forms. In general, the form of communication should be appropriate to the creative task and information being communicated. For example, in an improvised music performance, it would be inappropriate for participants to stop playing and start talking about why one made a particular musical decision in the previous bar. Often visual, sonic, haptic or even olfactory communication may be the most suitable, and the form of communication can influence the creative outcomes in subtle, non-obvious ways. For example, the smell of freshly baked bread or a spring meadow can trigger specific memories or support synaesthetic-like relationships to other sensory media (Ackerman 1990).

**Discussion**

**Challenges and Opportunities**

Different issues surround the idea of explainable models. A concern that has been raised in the XAI field is that explainable models are not accurate representations of the actual functionality of the system in some aspects of the feature space. According to (Rudin 2019) “an inaccurate (low-fidelity) explanation model limits trust in the explanation, and by extension, trust in the black box that it is trying to explain”. Although we believe that CC provides an outlet in which explanations do not have to be faithful to the intrinsic motivations/objectives of a system (as has been postulated through the notion of framing in CC), we need to be careful so as not to endanger the trust that co-creators, users and

---

audiences impose into these systems. For instance, the CC advertiser in our example may be involved in pitching the concept of the ad to the client. In the process it may provide intuitions behind the campaign but in doing so it should not take credit for aspects of it that were actually ideas of its co-creators (e.g. the idea of blending dice and teeth in the example). Another common challenge is that explainable systems have a social responsibility. This has been explored extensively in (Arrieta et al. 2020) where the concept of Responsible AI is introduced together with some practical principles (such as fairness, transparency, and privacy). For instance, one thing is for the CC advertiser in our example to refer to gambling as an analogy to endangered oral health, and another (questionable possibility) is to use it to encourage gambling. Finally, there must be clear limits about the extent of which a CC system may try to champion an idea; i.e. CC systems cannot disrupt or take control over a creative process. Imagine for instance the CC advertiser in our example not backing down about using the first image of the poker table even though its co-creator had already rejected it.

Despite these challenges, explainable models, as suggested here, may open up possibilities of new kinds of social interactions between CC systems and their users. Work from the social sciences has shown that non-human entities can take up active roles in social practices and that imposing concrete boundaries or definitions between what roles these can or cannot play only limits their potential. Strengers (Strengers 2019) illustrates this point by looking at Roomba riding, a trend that refers to how pets enjoy ‘riding’ a robotic vacuum cleaner called the Roomba, and the potential of this type of technology to become a pet entertainment device. Imagine for instance a CC collaborator that strongly suggests its human co-creator to stop working during the weekend to be with his/her family, or a human co-creator that opens up a co-creative collaboration with a CC system on social media, or a CC system that proposes its human co-creator to watch a film together in order to have a break and maybe get some inspiration. Through this, the idea is to emphasize that in order to identify potential relational roles and dynamics between humans and machines, it is important to not only consider humans as capable of performing a social role, but machines should also be considered as capable performers of such roles. Equipping CC systems with explainable capabilities may reveal a type of dynamics between humans and CC systems that is not possible with most current technologies.

**Conclusions and Future Work**

We have presented Explainable Computational Creativity (XCC), as a sub-field of XAI that is focused on the applicability of explainable models in the area of CC and how these can be used to open up bidirectional communication channels between CC systems and their users. We have outlined four design principles we believe are crucial for these types of models, namely mental models (individual representations of how things work), long-term memory (the capacity to store and access details of past experiences), argumentation (the ability to reason and support creative contributions), and exposing the creative process (revealing specific details about the operation of a system). We believe that this framework will benefit human-CC interactions by: i) enhancing their creativity (as a result of working together rather than in isolation), ii) creating more fruitful and productive partnerships (by establishing bidirectional communication channels where creative contributions can be examined), and iii) increasing the engagement of users with CC systems (by improving the flow of these collaborations and the overall perception of their outcomes).

The proposed approach opens up questions such as: How can CC systems explain their creative acts and how do people respond to them? What makes a good or poor explanation in the context of CC? What are the challenges that CC systems face when providing explanations? How can CC systems change their behaviour in response to feedback, new ideas or historical knowledge of the co-creators they are working with? How do we evaluate CC systems with explanatory capabilities? We have explored the surface of some of these questions but as mentioned before, domain specific instantiations would dictate different interactions. We plan to explore this as future work in the context of...
of music improvisation and composition, by extending the SpeakeSystem with explanatory capabilities.

References


On the Machine Condition and its Creative Expression

Simon Colton1,2, Alison Pease3, Christian Guckelsberger1,4, Jon McCormack2 and Teresa Llano2

1 Game AI Group, EECS, Queen Mary University of London, London, UK
2 SensiLab, Faculty of IT, Monash University, Melbourne, Australia
3 School of Computing, Dundee University, Dundee, UK
4 Finnish Center for Artificial Intelligence, Aalto University, Espoo, Finland

Abstract

The human condition can be characterised as the most essential characteristics, events and situations which describe human existence. We propose that a parallel discussion of the machine condition could improve public understanding of computational systems in general, and advance perception of creativity in computational creativity systems in particular. We present a framework for machines to creatively express their existence, sketch some aspects of the machine condition, and describe potential benefits of this approach.

Introduction

There are many reasons to engineer software that can act in an autonomously creative fashion in arts and science projects (Veale, Cardoso, and Pérez y Pérez 2019; Pérez y Pérez 2018). These include: enhancing human creativity through support tools; increasing the well-being of groups of people; public engagement around issues of AI; and bringing novel, interesting and important artefacts into the world. We explore here a less well-studied purpose for AI systems to create, namely for them to tell us about themselves. That is, machines have experiences which – while accepting that they are generally not considered to be alive – could be considered life experiences. The re-telling of these experiences through creative practice could be useful in human-computer interaction terms. Given the high complexities of the processing, data, physical presence and sensory inputs present in many computational systems, and the high impact they have on human society, giving machines a way of expressing themselves creatively might help people grasp difficult elements of our technological society. This may initially have only utilitarian value in clarifying how technology works, and people may not care about machine life experiences. However, it’s not impossible to think that people could become fascinated with such experiences, and may, in time, develop empathy with machines as they become more integrated into, and appreciated by, society.

Increasingly, automated systems make decisions about people’s lives without the people it affects understanding how they work. Artistic production in human cultures enables communication which helps people understand each other and make connections within and between communities. We suggest that AI systems could undertake artistic production for the purpose of explaining how the software and hardware functions at all levels, rather than – or in addition to – the other purposes given above. This suggestion fits within the explainable AI (XAI) movement (Arrieta et al. 2020) and the many initiatives to educate people about how technology works in general. However, we go further in suggesting that we define and understand aspects of the existence of machines, not just the decisions they make or the processing they perform, and we suggest a mechanism for this communication, namely expressive artistic production. Artistic products such as paintings, musical compositions, poems, games, etc., differ in accuracy when used to communicate ideas. We look here at supplementing other efforts in computer science public engagement via machines producing artworks which, due to their more abstracted nature, demand (human) interpretation of the ideas expressed. While this may decrease accuracy in the communication of machine existence, the interpretative effort required could mean that people gain understanding on their own terms, and the cognitive effort may help ideas to persist in their minds.

Our aim here is to suggest a context within which such expressive production could be carried out, and to provide a framework to guide the initial construction of creative AI systems for the communication of machine existence. Gaining inspiration from human creative expression, we note that the notion of the human condition provides a framework for some art production, as it addresses the most important aspects of human existence. We therefore propose a parallel notion of the machine condition, i.e., what it means to be a machine, as part of a framework for creative expression by computational systems. This parallel notion is achieved via an interpretation of the human condition as a set of high-level concepts capturing categories of events, and an understanding of the types of events in a person’s life which constitute major experiences. The framework also suggests a pipeline for creative production whereby an actual event prompts the production of an artefact which references higher level aspects of machine existence.

Augmenting the reasons for machines to behave creatively will affect the way in which we evaluate such systems. In the next section, we describe how various evaluation methods for creative machines, for various purposes, have influenced our thinking and led to the proposal here. We also place our work in the contexts of explainable AI (Arrieta et al. 2020) and communication in computational creativity, and suggest how generative systems could be
taken further than they are currently, in both respects.

In order to begin the discussion on what a parallel machine condition could be, we consider aspects of the human condition, including distinctions between active and contemplative life. We present a workable simplification of this broad topic, which can be used to initiate a framework for considering and artistically expressing the machine condition. We then present this framework by discussing a machine self and the nature of life experiences, some caveats, requirements and desirable properties for aspects of the machine condition. Following this, we present a first draft of the high-level concepts, categories of experience and individual experience types that could be considered part of the machine condition, and make some suggestions for how machines could express this creatively.

We argue that generative AI systems able to creatively express aspects of their existence, will drive forward computational creativity research, and this is a focus here. However, we believe that other, non-generative, machines could be enhanced to also produce artistic output describing their existence. In the conclusions section, we expand this idea, and argue that such communication could eventually become the most important reason for AI systems to be creative.

**Background**

The suggestion that computational creativity systems produce artistic works to convey to audiences aspects of machine existence fits into three contexts of previous work.

**Explainable AI**

In the field of explainable AI (XAI), the main aim is to provide explanations of how and why black box models reach particular solutions to a problem (Arrieta et al. 2020). In mainstream AI, the focus has been on making neural models interpretable, for instance by providing rationalisation about how certain features within a neural network react to certain inputs (Nguyen et al. 2017), as well as how elements of a network interact with each other (Zeiler and Fergus 2014). Research into the role of XAI in creative systems is in its infancy, but is rapidly gaining attention in how we approach the development of co-creative AI systems, for instance in mixed-initiative game design (Zhu et al. 2018); e.g., by aiding game designers in understanding input-output patterns through the use of a shared vocabulary between designers and machines (Guzdial et al. 2018).

We are less focused here on particular tasks and more on general education around technology, afforded through creative production; less on correctness of a particular explanation, and more on conveying the similarities and differences between human and machine existence. Most recently, however, a new area called Explainable Computational Creative (XCC) has been proposed which not only highlights the role of explanations to describe a creative act, but also as mechanisms to make creative systems active collaborators, e.g., by adopting the notions of argumentation and advocacy (Llano et al. 2020). Although XCC promotes the idea of augmenting explanations of creative systems with past experiences, these are related to how the system operated in previous occasions, rather than in aspects of existence as proposed here.

**Communicative Computational Creativity**

Our contribution here also relates to existing work addressing issues of communication in generative systems. The notion of creative AI systems employing framing (Cook et al. 2019) bridges this context and that of XAI. Charnley, Pease and Colton (2012) advocate building computational creativity systems able to communicate, through text, aspects of their motivations, processes and internal evaluation of the artefacts they produce in generative projects. Machines communicating aspects of their existence would add to the sophistication of their framing; but the term framing only applies to systems which add value to a separate artefact through description of their creative acts. It hence introduces a second artefact generation process to describe the main act. We go beyond framing here, in what information and concepts are conveyed, which systems could do this, and why. As expanded below, the systems addressed here include but, importantly, are not limited to those associated with artefact generation or creative problem solving.

Also in this context, natural language generation systems for producing text output such as stories and poems (Gervás 2009; Corneli et al. 2015) clearly have communicative purpose. However, such systems are usually designed to convey aspects of human life, not machine existence. (Saunders 2012) in contrast considers various means of enhancing machine creativity through communication in societies of autonomous artificial agents. We can distinguish between communication through the exchange of artefacts (Saunders and Gero 2001; Hantula and Linkola 2018), and through language alongside artefact production (Saunders 2011). Of central interest in the latter is how the ambiguity in language can contribute to variety in creative production. Both modes of communication have also been leveraged to shape the interaction between co-creative systems and their human partners (Kantosalo and Toivonen 2016). We do not focus on a specific mode of communication here, but rather on its object and originator: the system and its life experiences.

**Computational Creativity Purpose and Evaluation**

Our contribution also relates to computational creativity evaluation, in that it suggests a specific evaluation perspective. Computational creativity systems can be considered and evaluated from at least four different perspectives (Jordanous 2016): the product as a system’s output, the process as the way it operates, the press as environmental determinants of creativity, and the producer as the characteristics of the creative agent. Most existing work in computational creativity focuses on evaluating a system’s product or process (Ritchie 2019; Jordanous 2019), while our contribution here targets the producer. Some frameworks describe the producer in terms of the behaviours it is capable of. In the creativity tripod (Colton 2008) captures three necessary conditions for an artificial system to be perceived as creative by people: skillfulness, appreciation and imagination. (Jordanous 2016) notes that all three conditions allude to personal characteristics of the producer. The majority of existing frameworks describe the producer in terms of its functional components. For instance, the creative systems framework (Wiggins 2006), a formalisation of Boden’s
(1990) model of creativity, describes a system in terms of rules to validate and evaluate concepts, and rules to traverse a conceptual space. Ventura (2016) elaborates the necessary components for a creative AI system to exceed the threshold of being “merely generative”, and Ventura (2017) presents a blueprint of computational creativity systems as a practical guide to their construction.

A potent argument against the idea that a machine can be genuinely creative is that it lacks autonomy (Saunders 2012; McCormack, Gifford, and Hutchings 2019), specifically the intentional/mental autonomy typified by contemporary theory of mind (Boden 2010, Chapter 9). To address this, there has been much work on increasing the level of autonomy in creative AI systems to cover what it does (Jennings 2010; Linkola et al. 2017), and why it does these things (Guckelsberger, Salge, and Colton 2017), as well as how it frames its creative acts. However, to date, no system exists that exhibits the intentional autonomy that philosophers such as Boden argue is fundamental for human creative practices, and mechanisms to achieve it remain illusive.

While giving machines more artistic license through increased autonomy may improve the chances of people accepting them as creative, this may be counteracted by issues of authenticity (McCormack, Gifford, and Hutchings 2019). Colton, Pease and Saunders (2018) argue that AI systems which make artefacts about particularly human-centric issues, like childbirth, will naturally be seen as inauthentic. One suggestion for combating accusations of inauthenticity given by these authors is that creative AI systems record details of the interactions they have in their environment, so that these observations can be referenced in future creations in an authentic way. We pick up on that suggestion here, but take it further, to encompass any system recording aspects of its existence and using them later to convey aspects of how it operates. This will help the system appear more authentic and possibly more creative, and also improve public understanding of AI and technology in general.

The Human Condition

Many disciplines and movements offer reflections on the human condition as fundamental issues of human existence and the meaning of life. Much of philosophy is dedicated to finding meaning in life, and answering questions such as how we should live, what is human nature, and what society should look like. Answers are offered in moral philosophy, utopian visions of human society, the role of truth in philosophy, utopian visions of human society, the role of truth in human enquiry and even movements such as existentialism, which suggest that ultimately life is absurd with no meaning. There is much overlap between philosophical and literary perspectives on these themes. Psychological perspectives focus more on what we need as humans, what it is to have an identity, and our personal search for meaning. Religious teachings focus on aspects such as sin, morality, cycles of life, submission before God and how to live a prudent and mindful life. Other perspectives come from biology, anthropology, history, art movements and numerous other sources.

Of particular interest here are the thoughts of Hannah Arendt (1998), who picks up on the ancient characterisation of human life into active (vita activa) and contemplative (vita contemplativa) elements, and asserts that the relative concerns are different, but neither is the more important. If we focus on the active life, Arendt builds on the ideas of Kant, Marx and others, and distinguishes human labour, work and action, and charts their changes through Western history, affected by important world events.

To make progress, we take a relatively simplistic and constrained view of the human condition, specifically grounded in the vita activa of events that actually happen to people, rather than emotions they feel or thoughts they have. Our aim of describing a computational parallel, this seems appropriate, because events and actions do take place which involve machines in similar ways to those involving people, but the idea of machines feeling or thinking is controversial and adds extra complexity. Taking a hierarchical view, and starting at the highest level, we see the human condition as a small number of under-specified but important concepts such as growth, death, conflict, aspiration and love. Under each umbrella term, we can identify different categories for types of events to be placed into, albeit with overlap. For instance, the high-level concept of death covers event types which can be categorised into: grief, mourning, pain, loss, dying, etc. Individual events related to, say, dying, involve acts such as a murder or someone attending a funeral.

To ground the discussion, imagine someone writing a poem about a leaf falling from a tree, as an analogy with which to express their grief, given a particular event: the recent loss of a loved one. Here, the aspects of the human condition being expressed might include death and love at the highest level, and grief, loss and mourning at the second level. While readers of the poem may be aware of these notions and have personal experience thereof, the individual expression of them by the poet may be new and insightful. For instance, the poem may allude to (or guide the inference of) an analogy between the tree losing a leaf and the world losing a person. This analogy may offer something new to the reader, enabling them to understand the particular grief experienced by the poet, perhaps here highlighting personal loss as part of a global loss. We could speculate that a particular reader of the poem might believe it to be ‘shallow’ if the poem only communicated aspects of a leaf falling or a person dying, ‘deep’ if it communicated some aspects of the grief of the author, or the notion of grief in general, deeper still if it led the reader to think about the notion of death, and very deep, if the poem ultimately led the reader to consider what it means to be human.

The Machine Condition: Prerequisites

Recall that we are interested in the notion that a computational system could use creative production (of poems, visual art, games, stories, musical compositions, etc.) to communicate its experiences in a way that potentially encourages audience members to think about machine existence. We upgrade the term ‘experience’ to ‘life experience’ to convey an event of particular importance to the trajectory of existence of a particular person or machine. We take the position that an entity like a machine does not need to satisfy notions of being alive or conscious to have life experiences worthy of communication through creative expression.
In sketching out aspects of the machine condition, we ask the question: “What is it like to be a machine in the year 2020?” This mirrors the famous thought experiment asking “What is it like to be a bat?” in (Nagel 1974), which raised issues of consciousness in humanity. We note that bats evolved through natural selection, and their experiences are not as easily subjected to interrogation and experimentation as they are for machines, which were engineered, not evolved. Hence understanding machine existence may be less hindered by the difficulties proposed by Nagel, who suggested that, as we don’t have things like a bat body or a bat brain (nor sonar for “seeing”) we can never know what it is like to be a bat from subjective experience.

We adopt the structure imposed on the human condition above as a starting point for addressing the machine condition. That is, we determine some high-level concepts, categories of events and individual event types that could describe the most important aspects of the vita activa of machines. Before providing some initial suggestions to populate this structure, we first discuss some prerequisites about individuality and computational life experiences, air some caveats and describe some requirements and desirable properties for aspects of the machine condition. To limit our exploration, we propose a specific production pipeline whereby a machine has a life experience which fits into a category of events that could be used to portray some high-level aspect of machine existence. Details of the experience are captured at the time and used later in the creative production of an abstracted artwork which people could in principle interpret via contemplation of the machine condition. The artwork could be supplemented by a text which frames the creative act and provides additional understanding of the aspects of machine existence expressed in the artwork.

As an example, imagine a generative art program producing abstracted artworks on a laptop in a public space. People sit and watch pictures emerge on the laptop screen while the software records its internal state, network environment and aspects of its external environment (through a camera and microphone). When someone accidentally spills coffee on the laptop, this leads to the laptop suddenly shutting down. When the software is next run, its internal sensors highlight the physical environment has changed from a laptop to a desktop computer, and its code has been copied from a repository, with slight changes imposed. This life experience could be used later on by the system to portray one of a number of aspects of machine existence, expressed via an artwork and an accompanying piece of text.

Individual Life Experiences

It is important to address the notion of a machine self, i.e., what exactly we mean when we say a machine has expressed itself creatively? We restrict ourselves initially to consideration of software and hardware intertwined into what would normally be considered one system. Such a system might simply be software on a laptop computer changing pixels on a screen, or might be a desktop computer running software that controls a robotic arm. We further specify that the system should be able to record details of events that happen to it, in order to use them in creative production later on. Both the software and hardware component can change over time, but it is reasonable to think of it as still the same system, similar to how a particular person changes physically and environmentally over time, but never has a different self.

With the term life experience, we leave open the full range of experiences a system could have. At this stage, we specifically include (i) changes in the code and architecture of its software (ii) changes in its hardware (iii) changes in the data it processes (iv) changes in its computational and network environment (v) changes in its physical environment (vi) interactions with other software systems and (vii) interactions with people. We assume the software can record such experiences with similar or higher accuracy than a person.

Caveats, Requirements and Properties

It may be tempting to apply a mapping from aspects of the human condition onto aspects of machine experience. For instance, we could look at the notion of death in humanity and search for machine experiences such as the deletion of its code, onto which to map the notion of death. There are (at least) three difficulties with this approach. Firstly, aspects of machine existence which map nicely to human existence may not be particularly representative, e.g., entire deletion of code is not a particularly common thing to happen to machines, and certainly not for the type of systems we are focused on. Secondly, the analogy may not hold perfectly, and actually serve to confuse our understanding of machine existence rather than clarify it. For instance, the wholesale deletion of code is more akin to the science fiction idea of wiping a person’s brain, than the complex notion of human death. Thirdly, mapping the human condition onto software existence likely serves more the purpose of understanding humanity than increasing our understanding of machines.

We have to be equally wary of mapping actual machine life experiences onto aspects of the human condition, for the same reasons as above. A number of projects have demonstrated that software can be specifically engineered so that we can project an element of the human condition onto it. For instance, imagine a generative art system described by its author as ‘suffering’ when it senses that no-one is watching it. Such an exercise demonstrates only that it’s possible to engineer scenarios in which actual computational situations map onto aspects of the human condition. However, the situation is in part artificial, because the relevant life experiences (i.e., the event of sensing that no-one is watching) have been engineered entirely for the project. Such projects may encourage us to think about human suffering, but as there is no machine equivalent, they do little to increase our understanding of machines.

To make some progress, we can specify some initial requirements of the life experiences which could be used in creative expression by machines as follows:

- The experience actually happened to the machine as part of its processing in a non-cyclic way, i.e., for purposes other than harvesting experiences for creative expression.
- The experience is reasonably unusual and distinctive to day-to-day experiences that the machine has.
• The experience reasonably fits into one or more categories of high-level software existence that captures an important aspect of the machine condition – with some suggestions for such categories supplied below.

Over time, we expect these requirements to be relaxed, as new ways for machines to communicate through creativity emerge. In addition, we can suggest some desirable, but not necessary, qualities of machine life experiences, to help further narrow down the scope of creative expression exhibited by initial implementations. In particular, it may be sensible in the first instance to concentrate on experiences which highlight aspects of the machine condition which are most different to the human condition. This may entail that the communication of machine existence is less easily confused with a projection of humanity, and perhaps speed up understanding of machine existence overall, i.e., by tackling differences to us rather than similarities which might be easier to understand. Also, it might be sensible to concentrate on life experiences and aspects of the machine condition which are possible to express succinctly in a single artefact, so that any explanatory text (see below) is kept to a minimum.

Aspects of Machine Existence

We have identified five high-level under-defined concepts into which machine-centric events can be categorised, namely transience, learning, humanity, work and physicality. These are meant to be at the same level as the notions of death, growth, aspiration, conflict, love, etc., in the human condition, and each is expanded below in terms of the categorisation of events that they afford. These areas are given in no particular order, and there is much fluidity in the categorisation, i.e., some event types could be moved between areas, or should perhaps be considered in multiple categories, as is the case with event types for the human condition.

Transience

The changing nature of basically all aspects of the existence of a particular machine deserves to be considered as a particularly important notion in describing machine existence. In no particular order, transience of machine existence includes events which fit into categories such as network variation, changes in data, alterations to the machine’s physical and software components, changes in the local external (human) environment in which the machine is located, and global changes to the human world. Event types which might class as life experiences in these categories include: a substantial new module in a machine’s code; the changing of a robotic component; movement of the machine to a new venue; a new trend on a social media platform like Twitter, if the machine is processing data from this stream; and web pages it refers to, or an entire network, going offline.

Returning to the generative art system in the coffee-spill example, suppose it retrieves images to use as art materials by querying sites on the internet. Such processes are liable to change regularly, due to the everyday shift in the structure of the internet. In the case of major events, like a nameserver being unreachable, this could result in dramatically different times or routes for connections, which could affect how long creative processes take, what order different tasks are completed in or how far a search can be conducted. In addition to network structure changing, the content may change, especially when searching for data that is algorithmically curated. Searching for images about ‘coronavirus’ in early 2020 will result in very different results depending on the month the search is conducted, where in the world the originating connection is, and what other things have been previously searched for. While people may not even perceive these shifts, software may experience it very differently, particularly if the system evaluates the results of the search in order to make a decision. In general, if a machine could communicate the transient nature of the network it operates in, this may help people to understand unexpected or seemingly inconsistent outputs or decisions from it.

As software is developed, its codebase grows and contracts as code is added, refactored or removed, and it is common to use version control to manage the codebase and record how a project develops. Analogies with the human experience such as growth, development, evolution, learning and training don’t quite fit the experiences of a machine witnessing in short timescales its changing self. Recording and retelling events relating to its changing codebase and subsequent changes in its affordances offers opportunities for a machine to express the transient nature of its existence as an entity. Such changes are inaccessible to an observer, but deeply relevant to their experiences of the machine, since the code informs what the machine does, is, and can potentially achieve. If a machine can communicate aspects of the evolution of its codebase, this may help people understand the impact of bugs and how they lead to errors or delays, and/or help people be more patient with respect to missing features.

Learning

How a system comes into existence is something worthy of communicating through creative expression, as understanding the origins of a machine may help people comprehend how it works and the impact it has on their life. One major concept in this context, which acts as an umbrella to many aspects of the machine condition, is learning. Given the impact that one set of techniques in this area, deep learning, is currently having on society, and that these techniques are the main focus of XAI work, expressing elements of learning seems imperative. Categories of event types here could include: the training of a new machine learning model which forms part of a machine; a new dataset being used for the training; evaluating a trained model’s predictive accuracy or it’s value against some other set of measures; the running of such models for predictions, categorisations or generative tasks; and analysis of the results of such running. We can also include events which relate to the human programming of a machine, or the automatic generation of code for a machine, through, for instance, genetic programming techniques. Another set of events which might fit under the umbrella of learning include those related to physical memory (hard-drive or RAM) access, and differences in human and machine memory around permanence, ease of retrieval and representation could be expressed in relation to such events.

In the coffee-spill example, imagine the generative system
was augmented with a machine vision neural model able to
analyse the art images that it produced. The first running
of the machine vision model to analyse a generated image
would constitute a life experience of significant magnitude.
This could be conveyed through creative production in such
a way as to highlight some of the above general aspects of
machine learning, e.g., comparing how people see textures,
colours and shapes in an image, contrasting with the ma-
chine, which calculates numerical outputs from an artificial
neural network to analyse an image.

**Humanity**

The machine existence is decidedly within a world of hu-
man existence. Machines are often made by and for indi-
vidual people and human communities. They interact with
and influence people, but are not treated as equals, but rather
as employees for human employers. In general, software is
programmed/trained to be good at things that people want
or need to do, like play chess or detect spam email. In-
deed, there is an assumption in some quarters of AI research
that achieving human level and human-style abilities is the
only goal of the field. This all-pervasive enveloping of ma-
chine existence by humanity forms an umbrella concept to
add to the description of the machine condition. Some cate-
gories of events here include: human-computer interactions,
whether with programmers, users or audience members; de-
cision making, as machines make decisions that affect peo-
ple, and vice-versa; communication, where the aim is to
convey some information to a person, rather than to interact
with them; execution, capturing events where someone com-
mands a machine to do something; and agency, where initia-
tive in a scenario switches between human and machine.

Only in the last 50 years have people begun to have in-
teractions where another entity/species (i.e., computers) has
more cognitive abilities than it, albeit only in very specific
ways. Machines have interactions with a more intellectual
species (humanity) on a daily basis, and it may be instructive
for people to understand this from a machine’s perspective.
Moreover, how the inequality between human and machine
plays out in events that happen to a machine may be a good
target for creative expression, especially if machines begin
to be in charge in certain scenarios, i.e., machines begin to
employ people for their benefit, not the benefit of other
people. Some other issues which could be addressed include:
differences in aspects of mortality/immortality between hu-
mans and machines and the difficulty machines may have in
describing how they operate to people.

In the coffee-spill example, an event with a person led
to a life experience for the machine. This could be used
to express transiency as the software moved from laptop to
desktop, but could also be used to express the relationship
machines have with people, in particular the low level of
agency machines tend to have: no matter how much auton-
omy a machine is seemingly given, it can still be rendered
useless by someone, whether on purpose or by accident.

**Work**

Machines exist primarily to carry out work which leads to
changes in a physical and/or virtual environment and/or the
creation of new knowledge and artefacts via information
processing. If any one aspect of the machine condition most
captures the essence of machine existence, this is probably it
– machines tirelessly and endlessly carry out work of value
to human society. Event categories under the umbrella of
work could be associated with: how software operates, e.g.,
loops, conditionals, subroutines, etc.; data and how it is anal-
ysed and transformed; execution traces; energy consump-
tion; the degradation of hardware; evaluation, as people and
machines determine how good a piece of machine work is;
the nature, collation and storage of output; hierarchies of
control and responsibility in software, hardware and human
groups; and benefactors – whether people or other machines
– who consume the results of machine work.

In the coffee-spill example, the generative process is
clearly the work that the machine undertakes, and this is
stopped not by the software, but by an accident. One as-
pect of difference between human and machine existence
that could be expressed here is the notion of responsibility
– while it is the machine undertaking the work, it holds no
responsibility for whether the work gets done. In general,
the output of machine work tends to be tailored for human
consumption, but the processing and reasoning behind the
undertaking of a piece of work does not. If machines could
express other aspects of the work they do, this could reduce
frustration when software performs poorly, and lead to
improved computational thinking in the general public.

**Physicality**

Computational systems, even though they can be thought of
abstractly, and aspects such as code tend not to be thought of
in physical terms, do have a physical reality. This ranges
from a server quietly processing information in a dark base-
ment to a robot constructing a car in a factory. The nature of
this physicality and how machines both sense and affect the
physical world, is an inescapable part of the machine condi-
tion. Event categories here could be associated with: mea-
surements, such as weight, mass, volume, degrees of free-
dom, location, reach, speed and accuracy; presence, where
machines occupying physical space affects how people react
to them; upgrades, where better parts are substituted; sen-
sors and the nature of the data they record; the presence
of noise in sensors, actuators and the environment, how it af-
facts machine operations and how it can be dealt with; the
changes that physical operations have on the environment,
which could be permanent, e.g., welding one car part to an-
other, or temporary, e.g., displaying an image on a screen.

The generative system in the coffee-spill example could
use its camera to capture pictures of its surroundings as art
material for making art, and it could easily be attached to a
printer so that physical printouts of its work could be made.
This may increase the presence that the machine has, and
how people react to it, which could be expressed in later art-
works. In general, machines sense and affect the world dif-
ferently to people, and if they can express these differences,
we will probably gain a deeper understanding of the world
around us. Machine physicality expressed through creative
production may also help us come to terms with the fact that
we are indeed sharing the world with machines.
Creative Expression of Machine Existence

The pipeline mentioned above suggests a straightforward way in which a machine could express aspects of its existence, and that of machines in general, prompted by a life experience it had. The construction of an artefact, whether a piece of music, visual art, poem, game or otherwise, will naturally draw on the many generative techniques and methodologies developed in computational creativity research and elsewhere. We offer here some general thoughts which could supplement this wealth of knowledge.

The two events in the coffee-spill running example above (the spill and the movement of the software from laptop to desktop), could be used to tailor the generative software’s art production techniques in order to convey to people aspects of the machine condition related to transiency (with the software being moved and changed), humanity (how the machine inhabits the world with clumsy people who do unpredictable things) and possibly work (how its work can be stopped by an outside force completely beyond its control). The choice of what to express will be key, and it may be sensible to limit the scope to one aspect per artefact produced, to increase clarity. The tailoring of the generative process could be achieved by the software altering its generative parameters, workflow and/or source material, but could equally be achieved by the addition of new code by the software itself or by a programmer.

One possibility would be to take a literal approach and juxtapose some imagery of a coffee spill with a picture of a laptop and a desktop computer. However, this might be criticised as not being very deep, given that it does not offer much opportunity for interpretation or lead to insights into the nature of machine existence. Another approach may be to allude in the artwork to an idea from human culture, for instance the science fiction meme of one person’s mind being trapped inside the body of another, or a particular work from human culture, for instance Kafka’s Metamorphosis. Given that the artworks are intended for human audiences, references to ideas and artefacts from human culture may help to communicate aspects of software existence. A third approach would be to make analogies to human existence. While we advocate focusing on events which happen to machines and not drawing strict analogies at high levels (such as death being the same as code deletion), it makes sense to reference human life in particular generated artworks. This is because the looseness of the analogy may encourage audience members to interpret the message personally and possibly grasp the ideas expressed on their own terms. It would be important here, though, to remember that the aim is to communicate aspects of machine existence through reference to human life, not the other way around.

While many people in technological circles may understand how machines operate and the environments they work in, it is fair to say that the general public do not fully know what software is and does and how it controls machines. As mentioned above, the idea of software framing its work with explanatory text about its motivations, processes and outputs could help people to project notions of creativity onto the software. Given the differences between machine and human existence, makes sense to add to the potential focal points for framing some details of the individual events (e.g., coffee spill/restart) and bigger picture concepts (e.g., transience, physicality) that influenced the conception, production and assessment of a particular artefact.

Ultimately, people consuming computer-generated artefacts will have to put in some effort to think about the concepts raised in their own terms, if they want to understand more about machine existence. There will be many aspects of this that people will find difficult to grasp, and these could be targets where more explicit framing could be employed, to supplement more abstract presentations in the generated artefacts. These difficulties will likely include issues of scale, both in terms of volume of data and rapid change. Other difficulties might involve the complexity of the processing undertaken by a machine, involving thousands of lines of code, or the black-box homogeneity of systems like artificial neural networks. Further difficulties may include counter-intuitive notions and other-worldliness which make comprehension of how a machine works hard, e.g., as in quantum computing. The opposite of other-worldliness might also cause difficulties, i.e., false equalities resulting in the projecting of notions of humanity onto machines which cause errors in understanding, e.g., thinking of a robotic arm as being the same as a human arm. In light of these difficulties, it may be necessary for the generative aspects of a machine expressing its existence to employ a model of general human comprehension of machine existence, and use this to determine which aspects of the machine condition to portray in the artefact produced and the framing text.

Conclusions and Future Work

We have argued the case for why machines able to express aspects of their existence through creative production could offer a new purpose for computational creativity systems, and drive the field forward. We provided a simplified account of the human condition which suggested a structure for a parallel understanding of the machine condition. After describing some prerequisites, we discussed event types and categories thereof which broadly fit into five areas related to machine existence, namely transience, learning, humanity, work and physicality. Finally, we provided some thoughts on creative expression of the machine condition by computational creativity systems, and overall we hope that this work provides an initial framework to refer to when implementing a creative AI system able to express through its generated artefacts what it is like to be it.

The rise of generative deep learning and other techniques in the last decade has meant that automatically producing high quality artistic artefacts in volume is rapidly becoming less of an issue in computational creativity. This provides an opportunity to concentrate on the societal issues in the field, in particular the questions of why we want machines to be creative, and in which cultural contexts this would be appropriate. Suggesting that machines be creative in order to communicate about their experiences opens up new avenues for research with respect to these existential issues. We wouldn’t expect or want a ticket machine in a railway station to draw a picture or pen a poem to explain to us how the network is down and that this reflects the transiency of...
machine existence. However, it might be beneficial for an household robot to provide an accessible account of an incident or decision in some artistic form.

We ask children to draw pictures to express difficult events that occurred, and these can be enlightening, taken together with more direct communicative approaches like interviews and therapy sessions. The same could be true for machines, especially if the artefacts produced make us think about an aspect of the machine’s operation that we previously didn’t understand. As machine creativity rises, there may be a tendency for human creative activity to be prized in society, perhaps for reasons of community and authenticity. If this is the case, and we find that machine communication through expressive art is successful in the ways described above, it is not impossible to think that computational creativity research may ultimately find most benefit in communicating the machine condition, with the value of generated artefacts a welcome side-effect.

There are many questions about the machine condition left unanswered, and many ways in which this idea could be carried forwards. Firstly, we could investigate the ways (if any) in which certain groups of people care about what it’s like to be a machine. Improved understanding of how machines operate and affect society are general benefits arising from detailing and understanding the machine condition. However, there may be more specific reasons that people want to think of machines as companions, carers and collaborators, where an understanding of machine existence may help. Secondly, we could be more concrete about what a machine experience is, expanding past event-based experience and discussing how a machine would know whether it was having a life experience or not, and what other types of experience it might have. Thirdly, we could draw on semiotics and communication theory as qualitative and quantitative frameworks to distinguish what a machine tries to tell us about its existence, how this is signified through different artefacts, and how effectively and efficiently it is communicated. Finally, we could address the moral issues thrown up by people imposing notions of a machine condition, rather than them coming from the machines themselves. Indeed, one long-term milestone might be machines which define their own condition and surprise us by describing aspects we hadn’t previously thought of.

Acknowledgments
We thank the anonymous reviewers for their insightful feedback. CG was funded by the Academy of Finland Flagship programme “Finnish Center for Artificial Intelligence” (FCAI) and by EPSRC grant EP/L015846/1 (IGGI CDT).

References
Boden, M. 2010. Creativity & Art: Three Roads to Surprise. OUP.


Gervás, P. 2009. Computational Approaches to Storytelling and Creativity. AI Magazine 30(3).


Pérez y Pérez, Rafael. 2018. The Computational Creativity Continuum. In Proc. ICCC.


A Study on Reproducibility in Computational Creativity Research

Martin Marzidovšek
Jožef Stefan International Postgraduate School
Jožef Stefan Institute
1000 Ljubljana, Slovenia
martin.marzidovsek@ijs.si

Senja Pollak
Jožef Stefan Institute
1000 Ljubljana, Slovenia
senja.pollak@ijs.si

The importance of reproducibility has been recognised as one of the crucial elements of research especially in life sciences, but also in other disciplines, e.g. medicine, social sciences, natural language processing who have acknowledged the reproducibility crisis. In this paper, we discuss the issue of reproducibility in the field of computational creativity. We present the findings from an indirect reproducibility study that assesses the transparency of available information to allow reproducibility. It does so through the analysis of articles published in ICCC proceedings, which has shown that most computational creativity (CC) publications until now have provided very little with regard to documentation and resources needed for reproducibility and replicability of the published results. By reviewing best practices from the broader scientific community and considering the particularities of CC recommendations are put forward that will hopefully inspire the creation of standards and practices related to reproducibility and improve the replicability of CC research within the CC community and beyond.

Introduction

The goal of science is acquiring knowledge about the world for which it depends on the “ability of the scientific community to scrutinize scientific claims and to gain confidence over time in results and inferences that have stood up to repeated testing” (CRRS1 2019). Science is thus an iterative but also collaborative process that advances (more efficiently) when researchers can build upon others’ work, reproduce and reuse their results. For this to work scientists should as much as possible share data, methods and results, report uncertainties about their findings, and share not only positive, but also negative results.

While it is hard to claim that science is fully "objective", building trust in science requires standards and procedures that are accepted by scientific community. Karl R. Popper (1934 [2002]) claimed that “the objectivity of scientific statements lies in the fact that they can be inter-subjectively tested”. So for him “objectivity” did not depend so much on the correspondence of a scientific claim to facts as it did on its verifiability: whether it can be tested and put under rational scrutiny. Over the years, common scientific practices and standards have evolved such as reproducibility, which many see as a cornerstone of the scientific method since it enables the scientific community to confirm or refute research results, but also allow for reusability of prior research.

Reproducibility is not by itself concerned with the correctness of the results or the process. More importantly, as long as research is reproducible bugs in the code and flawed methodology can become transparent for other researchers who can improve the original work and so scientific critique and progress can be made. Reproducibility thus stands for providing a complete and unambiguous description of the entire process from raw data to the final results. “[…] When a researcher transparently reports a study and makes available the underlying digital artefacts, such as data and code, the results should be computationally reproducible” (CRRS 2019).

While reproducibility is concerned with obtaining quantitative scientific results by independent scientists using the original datasets and methods, replicability is as important, since it concerns validation of specific findings with other datasets and implementations of the original methods (Stoddent 2014, Branco et al. 2017). Not less important is reusability referring to the capacity to reutilize a novel component from the original research in another system even when insufficient resources are provided to allow for reproduction of experiments.

In the recent years, however, concerns about reproducibility and replicability – widely considered as hallmarks of good science – have increased in the scientific community and critical articles in high-profile mainstream media even spoke of a reproducibility “crisis” (Nature 2016, The Economist 2013, Branco et al. 2017). The quality of empirical results has been questioned and importance of reproducibility highlighted in many fields, such as (bio)medicine (e.g. Prinz et al. 2011; Ioannidis 2005), neuroscience (Button et

---

1 Committee on Reproducibility and Replicability in Science (US National Science Foundation).
al. 2013), economics (Camerer et al. 2016), language technologies (Branco et al. 2017) etc.

The ability and effort required from other researchers to replicate experiments and explore variations depends heavily on the information provided when the original work was published (Gundersen et al. 2018). We believe that, even if not applicable to all types of papers, for majority of scientific papers which describe empirical studies one can claim that if published research is not reproducible, it is of much less value.

By taking into account the specificities of Computational Creativity (CC) field this paper elaborates on the importance for the CC community to share with their published research complete and sufficient documentation about the used digital artefacts (e.g. datasets, code), methods and complete results to facilitate reproduction and replication.

Relevance of Reproducibility for the CC Field

As a sub-field of Artificial Intelligence research, Computational Creativity (CC) is the philosophy, science and engineering of computational systems which, by taking on particular responsibilities, exhibit behaviours that unbiased observers would deem to be creative. (Wiggins, 2006; Colton and Wiggins, 2012). As mentioned by Colton and Wiggins (2012) the methodological requirements for evaluation are crucial for the field, where they emphasise the involvement of unbiased observers in fairly judging the behaviours exhibited by CC systems.

While in CC there are different types of papers and contribution made, we believe that at least for technical and system presentation papers, the reproducibility is very important underlying aspect, which would allow for better and quicker development of novel systems in the field, more transparent evaluation of different systems, easier update of the research by novel researchers in the field, and better integration of previous work in novel systems. One could question, whether in the field where creativity and novelty is at the core, there is place for reproducibility, but we argue that striving for scientific research standards could be beneficial to an evolving field such as CC and could help its progress. As has been shown already by Platt (1964) particular scientific fields move more rapidly because they adopt systematic research methods.

Thus, in similar ways as other scientific fields, CC could try to rely more on reproducible experiments to validate research results, new discoveries and practices. We propose that the CC community should, whenever possible, strive to facilitate reproducible and replicable research by adequate experimental design and methods as well as clear and complete documentation in the publications. We advise to establish practical and pragmatic practices on how to document the scientific methods and resources so that reproducibility and replicability of CC research results is feasible in practice. In addition, sharing the code and developed tools would allow easier introduction to the field of new researchers and would be beneficial for the promotion of the field across disciplines and communities.

This paper presents the findings from an indirect reproducibility study that assesses the transparency of available information in published CC research to allow reproducibility. As will be shown in the analysis of International Conference on Computational Creativity (ICCC) proceedings most CC publications until now have provided very little in regard to documentation and resources that would allow, let alone facilitate, reproducibility and replicability.

It is crucial for CC as a scientific field to understand the reasons for that and to address these issues with the goal of improving transparency of the published research. While this paper can’t fully answer this question it aims to raise awareness of these issues among the CC community and inspire further inquiries and work that would bring about better reproducibility and replicability practices that would subsequently lead to fewer studies that do not reproduce or replicate.

Background and Related Work

In response to the already mentioned concerns about reproducibility of science by both the scientific and mainstream media the US Congress initiated in 2017 an assessment that resulted in a comprehensive 2019 report Reproducibility and Replicability in Science prepared by the Committee on Reproducibility and Replicability in Science (CRRS). Because the terms reproducibility and replicability are sometimes used interchangeably or have different and even conflicting meanings depending on the scientific field we use the definitions proposed by CRRS (2019):

“Reproducibility is obtaining consistent results using the same input data; computational steps, methods, and code; and conditions of analysis.”

“Replicability is obtaining consistent results across studies aimed at answering the same scientific question, each of which has obtained its own data.”

The same Committee further specifies that reproducibility involves the original data and code, while replicability involves new data collection to test for consistency with previous results of a similar study.

From a slightly different angle: “replicability or repeatability is a property of an experiment: the ability to repeat—or not—the experiment described in a study” (Cohen et al. 2018). Reproducibility, on the other hand, is a “property of the outcomes of an experiment: arriving—or not at the same conclusions, findings, or values” (ibid.).

In addition to the above two terms we point also to reusability of the code. So when complete experimental resources necessary for reproduction or replication are not available, we believe that reuse of the accessible components can still be beneficial as it allows the community to collaborate and develop further as well as trigger more interest from the neighbouring scientific and engineering domains.
Results from Reproducibility Studies

As noted by Peng (2016), reproducibility which initially sounds like a trivial task has shown that it’s not always easy to achieve. The CRRS reviewed a collection of reproducibility studies across a variety of scientific fields and found that “[…] systematic efforts to reproduce have failed in more than one-half of attempts made, mainly due to insufficient detail on digital artifacts, such as data, code, and computational workflow” (CRRS 2019).

The famous 2016 survey “Is There a Reproducibility Crisis?” by Nature assessed ten factors that turned out to majorly contribute to irreproducible research and came to similar conclusions. Among the factors were: selective reporting, methods and code unavailable, poor experimental design, pressure to publish, low statistical power, not replicated enough in original lab, raw data not available from original lab, and others (Nature 2016).

A study on reproducibility in artificial intelligence (AI) sampled 400 papers from the AAAI and IJCAI conferences and found that computational AI research was not documented systematically and with enough information to support reproducibility (Gundersen and Kjensmo 2018).

Similarly in a recent study Repar et al. (2019) analysed several influential papers on bilingual terminology extraction (a field of NLP) from the past 25 years where they assessed the dataset, code and tool availability for the purpose of reproducibility and replicability. A surprising observation was that not one from the sampled papers made experiment code available and only a few provided links to tools where experiments were conducted. This severely hinders replicability. Repar et al. furthermore attempted to replicate one of the analysed papers (Aker et al. 2013) and despite closely following the original paper they obtained significantly worse results than the paper’s authors.

Sources of Irreproducibility and Irreproducibility

The above studies exemplify how difficult of a task it is to reproduce or replicate research that lacks experiment code, datasets and complete information about the implementation. The CRRS (2019) also claims that the greatest barriers to reproducibility are inadequate recordkeeping and non-transparent reporting. It thus follows that efforts to encourage more transparency in scientific publications would be beneficial.

According to CRRS (2019) there are a number of factors that make reproducibility of published research so difficult to achieve. In addition to missing access to non-public data and code the Committee also mentions inadequate record keeping (steps followed), non-transparent reporting, obsolescence of the digital artefacts, etc.

There is an important “conflict” of incentives between researchers who conducted the initial study and the independent researchers who attempt to reproduce the results. As Gunberrger et al. (2017) have argued: “independent researchers trust an empirical study’s results increasingly with the amount of documentation that is shared with them, while the effort to reproduce the results increases when the amount of documentation is reduced. […] On the other hand, the effort to document the research increases for the original researchers with the amount of documentation that needs to be shared, while the generality of the method is increased if independent researchers reproduce the results given less documentation”.

All research should be reproducible but it can be expected that not all research will replicate due the inherent risks of statistical procedures, researchers’ mistakes and biases (e.g. selection of methods that confirm the desired hypothesis, or selecting the hypothesis only after seeing the data, splitting data into subsets that lead to desired results, etc.). Consequentially preregistration is becoming more and more an accepted norm to deal with these problems. It requires researchers to register all relevant aspects and information about the scientific study (data collection, hypotheses, methods used, etc.) before they start with the research.

Reproducibility and Replicability Best Practices

The CRRS issued several recommendations for ways on how researchers, academic institutions, journals, and funders should help strengthen rigor and transparency in order to improve the reproducibility and replicability of scientific research. One of the most important recommendations for our analysis states (recommendation 4-1): “To help ensure the reproducibility of computational results, researchers should convey clear, specific, and complete information about any computational methods and data products that support their published results in order to enable other researchers to repeat the analysis, unless such information is restricted by non-public data policies. That information should include the data, study methods, and computational environment” (CRRS 2019).

As proposed by Repar et al. (2019) availability of datasets is an essential prerequisite for successful replication, while having access to original code greatly increases the ease or reproducibility and replicability experiments. In terms of record keeping a full compendium of artefacts is required from the original researcher. The CRRS specifies that the “computational details that need to be captured and shared for reproducible research include data, code, parameters, computational environment, and computational workflow” (CRRS 2019).

The CCRS report states that even when a project’s data are publicly available the analytical methods described by authors in scientific papers often lack sufficient guidance to reproduce the results. As also noted by Repar et al. “[…] even code itself is sometimes not enough without additional implementation notes and information on the operating systems and software used.” The recommendation 6-1 by the CRRS (2019) addresses this problem: “All researchers should include clear, specific, and complete description of how the reported result was reached.” The Committee further specifies, which details should be included (e.g. clear description of all methods used, data management, discussions of uncertainty, etc.).

The modern standard for reproducibility of research is to use computer code for everything. Meaning, it should be
avoided to do even minimal changes to the data (pre-processing), generation of visualisations and so on manually as it can introduce errors and such workflows are mostly not recorded and thus undetectable to others. Computer code, on the other hand is way less ambiguous and there is less room for misinterpretation.

Of course not all researchers have high computer programming skills and there are other ways how they can share or execute reproduction and replication experiments. Instead of, or better, in addition to sharing experiment source code a web tool (e.g. CloudFlows\footnote{\url{http://clowdfloows.org}}) can be used to replicate the experiment and enable others – especially those with less or no programming skills – to reproduce their results. “Availability of a tool or application (online or offline) where experiments can be conducted eases reproducibility and replicability, but also enables the reusability of results by a larger community” (Repar et al. 2019).

Journals and scientific societies have an especially important role to play in improving the reproducibility of research output. As is suggested in recommendation 6-7 CRRS (2019): “Journals and scientific societies requesting submissions for conferences should disclose their policies relevant to achieving reproducibility and replicability.” The CRRS encourages these entities to set and implement desired standards of reproducibility and replicability and make this one of their priorities. It also proposes the adoption of policies to reduce the likelihood of non-replicability with specific measures.

At Nature authors when “submitting manuscripts to Nature journals would need to complete a checklist addressing key factors underlying irreproducibility for reviewers and editors to assess during peer review […]. Nature’s checklist was designed, in part, to make selective reporting more transparent. Authors are asked to state whether experimental findings have been replicated in the laboratory, whether and how they calculated appropriate sample size“ (Nature 2018a).

Gundersen et al. (2017) points out that reproducibility is concomitant with open science, which involves sharing data, software, and other science resources in public repositories using permissive licenses. He also notes the FAIR Guiding Principles for Scientific Data Management and Stewardship, which are increasingly associated with open science and ensure that science resources have the necessary metadata to make them findable, accessible, interoperable, and reusable (Wilkinson et al. 2016).

Connected to open science is also modern digital scholarship that promotes credit to scientists who document and share their research products through citations of datasets, software, and innovative contributions to the scientific enterprise (Gundersen et al. 2017).

**Reproducibility in CC Community**

CC can be seen as a field of science with diverse research and engineering methods and output. It is thus not always appropriate to draw direct comparisons with other scientific research areas. Best practices from physics or life sciences are not necessarily transferable as the results and evaluations of scientific inquiries are more uniform and adapted to that particular field.

Nonetheless, CC should as far as appropriate compare its reproducibility and replicability practices to standards established in the general scientific domain and best practices from similar disciplines such as artificial intelligence. This is crucial so CC can stay in touch with reproducibility trends and will drive it to maintain the highest possible scientific standards so it can strengthen its status as a scientific discipline.

Computational creativity research as it is now defined and centred around the Association for Computational Creativity is still a relatively new field. In 1999 the yearly International Joint Workshops on Computational Creativity started but an even wider global research community began to evolve with ICCC conferences the first of which was held in Lisbon 2010. With an international conference and increased research activity CC as a field of research started to mature and establish its place in the broader scientific community.

As has been explained earlier the cornerstone of science is verification – the process by which scientists confirm the validity of new findings or discoveries by repeating the research that produced it (CRRS 2019). This has lead us to reproducibility and replicability as practices ensuring that this is possible and our central goal of this paper – to review these practices among the CC community.

To get a bird’s-eye-view about the state of reproducibility and replicability of the published research at the past ICCC conferences we analysed what we consider to be the centre-piece of modern reproducibility and replicability practice – the sharing of experiment source code, datasets and other relevant resources for the published research by means of public online repositories. Nowadays, there is a plethora of repositories for sharing datasets (see e.g. CLARIN initiative), and code (e.g. GitHub, GitLab).

We decided to measure GitHub links as a simple indicator to give an approximation about the trend and frequency of referencing digital repositories in CC articles published in ten years of ICCC proceedings (2010-2019). While acknowledging that GitHub by no means is the only possible way to share reproducibility related documentation it has since its launch in 2008 become the largest host of source code in the world and is thus a good candidate to give a hint on the practice of sharing digital artefacts such as source code and datasets in digital repositories within the CC community.

As can be seen in Graph 1, the GitHub links only slowly begin to appear in 2015, which is the year when GitHub already became the largest public online repository in the world, and a relative increase can be observed in 2019. However, as will be shown in the more detailed analysis of
the ICCC proceedings of 2019 the actual links hosting resources necessary to reproduce the authors’ published research are even in 2019 still quite rare – many of the included GitHub links point to third party resources used such as libraries, related source code, etc.

![Graph 1: Trendline of GitHub Links in ICCC Proceedings 2010-2019.](image)

Analysis

In order to get a better understanding about where the CC field is now with regard to reproducibility and replicability of research, we decided to analyse closer the CC articles published in the Proceedings of ICCC 2019 (Grace et al. 2019), which offers a good overview on where the field currently stands. It is important to note that the conference in 2019 marked the 10th anniversary of ICCCs, which signals that the CC as a field of research is maturing and that research practices and publishing standards in this scientific community are by now sufficiently established.

We have selected 34 articles from the proceeding (ibid.) that fall in the category of technical and system and resource description. The identification of the category has not been clear-cut as the category is not stated in the proceedings. The sample tried to include all of the articles that use empirical research or engineering approaches, where verification and thus reproducibility and replicability should be possible. We have omitted all of the articles from the creative submissions, those proposing theoretical frameworks, and methodologies.

As there are no standards yet regarding reproducibility and replicability in the CC community we relied on best practices in the wider scientific community presented earlier. The analysis of the sampled papers does not intend to expose, which tick-boxes a certain article failed to satisfy but rather to give an indication on which of the important digital artefacts are currently being shared in the CC community’s research papers.

The selected articles have been analysed for four indicators: raw data, source code, application or online tool, and complete results. These represent the minimum standards for reproducibility and replicability of scientific research that we consider crucial for the progress of CC as a field of scientific research. They are also easy to identify and are likely to be considered indisputably beneficial for the progress of open science and research by the CC community. However, deeper analysis of the factors relevant for reproducibility residing in the content of the articles is needed once more clear standards and policies by the ACC are established.

In the four selected indicators we were looking for whether the particular digital artefact has been shared by the author via a link directing to a digital repository or website that included complete or sufficient resources that would allow reproduction and replication of the study. This is in line with the already mentioned recommendation by CRSS (2019) stating that “researchers should convey clear, specific and complete information about any computational methods. […] That information should include the data, study methods, and computational environment.”

No special software has been used for identifying the selected indicators, which have been searched and analysed manually from the pdf version of Proceedings of ICCC 2019 (the same approach has been used to count GitHub links in ICCC proceedings (2010-2019)).

Findings

The analysis of the 34 sample papers from the ICCC 2019 proceedings (Grace et al. 2019) shows that only a minority of articles that have been accepted to the ICCC 2019 include the most crucial elements needed for reproducibility and replicability. As can be seen on Graph 2 only 20.6% of the analysed articles shared the dataset, 17.6% shared the source code, 14.7% provided links to web tools or applications, and only 8.8% shared complete results.

![Graph 2: Percentage of Papers Sharing Reproducibility Resources (Indicators).](image)

Only 32.4% of papers provided at least one of the analysed digital artefacts, which means that more than two thirds of the published papers did not provide any kind of resource that is indispensable for the reproduction or replication of the published research. This raises the question why authors don’t provide access to artefacts related to their research: is it a lack of concern, time, or will for the extra
effort needed to prepare the most basic resources or an intentional decision of keeping the research for themselves for various reasons?

While most papers offer at least some sort of description and sometimes mention the source of the dataset used for their experiments this does not suffice to allow others to reproduce their work. Also providing links to home pages from where the data has been obtained is not really useful for replication as the exact dataset would be needed.

A few papers mentioned or provided links to third party code that has been used or adapted for their research (e.g. libraries of specific components) this did not match the requirements of our source code indicator as only the complete experiment code that was used for the published research allows it to be reproduced and replicated.

In general it seems to be the prevalent praxis that authors in their papers include only samples of the data, code, pseudocode, results or screenshots from their web applications. This does not permit others to reproduce or replicate the results and to allow new members of the community or researchers from other scientific communities to be able to develop upon previous research.

One of the few provided links to web applications presented in some of the papers and one link to results were broken. In the first case we counted the link as being provided as a working link could be found in the shared GitHub repository. As more links will become obsolete with time ways to make access to resources permanent should be sought.

There have also been cases where functioning web platforms that resulted from the research are mentioned but no links are provided. If the research output is proprietary and the source code or the applications built are not intended for the public use it would be appropriate to clearly state this.

Discussion and Recommendations

We recognise that the predominant interest in CC is for the final research output (e.g. creative systems and artifacts) and that because of this documentation of other aspects of the CC research can fall short, which could on the long-term have negative effects for the CC field. It is thus important to draw attention to the described practices of reproducibility and replicability of scientific research that will allow CC to develop more efficiently.

CC researchers should follow best practices from other scientific disciplines and “describe methods and data in a clear, accurate and complete way” (CRRS 2019). In addition accessibility of complete results is especially important for the now predominant cases where source code and tools are not available together with the input dataset. As has been argued by Colton and Wiggins (2012), “in many projects, the output is carefully scrutinised by the program’s author, and only the best examples are shown to audiences, or used as exemplars in research papers, etc.” They refer to the curation coefficient as a means to understand the performance of a particular creative system. In this regard the availability of complete results give a better sense on the representability of the few output examples that are usually included in the papers.

Publishing and research practices that were discussed earlier and are presented in our recommendations can improve reproducibility and replicability in the CC research but they clearly require additional efforts from the authors particularly as they incorporate them in their work habits. But there are also direct benefits to the authors. Gundersen et al. (2017) state ten benefits, among them: practice open science and reproducible research; receiving credit for all your research products (by citing software, datasets, and other products); increase the number of citations to your publications (well-documented articles receive more citations); improved chances of being funded; improved management of your research assets, etc.

The benefits to the CC community are clear: maintaining repositories for CC code, tools and data, would make the field more attractive to young researchers, facilitate teaching of CC and allow for fruitful exchange of results between the neighbouring fields (e.g. NLP and CC).

What follows is a list of some general recommendations relating to data repositories, source code, the presentation of results, and more that could inspire standards for facilitating reproducibility.

Reproducibility and Replicability Recommendations for the CC Community (Researchers and Institutions):

1. The complete data, source code and results should be findable and accessible via shared open repositories (e.g. GitHub, GitLab, etc.); while informative, samples of the data and source code, pseudocode, tool screenshots, etc. in the paper are not enough.
2. In case of dataset restrictions (e.g. privacy, intellectual property constrains) or other valid reasons for not sharing digital resources, this should be explained. When feasible, permissions should be arranged for reproducibility purposes.
3. Provide description and documentation of the experimental design, hardware and software used, and a digital record of the workflow (e.g. data selection and manipulation, parameters, results at different stages, etc.). Use computer code for every step in the experiment as it provides a clearer and less unambiguous record of all steps (e.g. avoid manual data manipulation that can introduce errors and is undetectable).
4. Use integrated analyses and reports that combine code with data manipulation and visualisations (e.g. Jupyter notebook).
5. Strive for at least one replication by e.g. colleagues before submission for publication, which will expose weaknesses in the documentation.
6. ACC should promote clear transparency requirements through standards and policies for reproducibility and replicability.
7. ICCCs could introduce a reproducibility and replicability submission checklist.
8. ACC could promote and publish more replication studies (e.g. as a special paper type in the ICCC call for papers).
9. ACC could promote and support preregistration before the start of CC research to avoid replicability pitfalls (e.g. researcher’s biases, etc.).
10. Make proper use of statistical methods; estimate and report uncertainties in results.
11. Use persistent links (PURL) for all shared resources or have associated digital object identifiers (DOI) so the shared resource is findable and available permanently.
12. ACC could establish an open online repository for research output submitted to ICCCs so it would be accessible on one place for the CC community and other interested parties.
13. The ACC and ICCC’s should promote the use of web tools in research (e.g. CloudFlows) that would make reproducibility simpler and bring CC research results closer to people with less programming skills.
14. All members of the CC community should as much as possible promote, practice and support open science, the FAIR principles and digital scholarship.

Other relevant recommendations for data, source code, experiments and workflows, digital records and more can be found among others in Gunderen et al. (2017), CRRS (2019) and Repar et al. (2019).

Conclusions and future work

We have tried to argue that the CC community needs to recognize the benefits of reproducible science. We believe the CC community should strive for a more open and collaborative research by sharing, verifying and building upon each other’s research. Because of this CC researchers, publishers and institutions should strive to adopt better reproducibility and replicability practices and ensure that its research is transparent and well documented to make reproducibility as feasible as possible in practice.

The present analysis of the factors for reproducibility and replicability in CC research has only looked at the most basic requirements. As potential future studies might show any attempt at reproducing CC research in practice will likely need more detailed information and digital record of the entire experiment: data manipulation, parameter settings, workflow, software, interim and complete final results, etc.

The findings of this indirect reproducibility study open the question on how the necessity to reproduce and replicate scientific results is perceived by the CC community. What are the reasons for the relatively low attention given to these aspects?

In future work, we plan to analyse a larger sample of the archives, and complement the research with a survey and interviews, which would allow for understanding of the researchers’ attitudes towards this topic. We strongly believe that encouraging the community to enable reproducible experiments by sharing the resources and tools is of crucial importance and can drive CC as a scientific field even further. It will increase visibility and position it along other scientific strands that build upon the foundations of falsifiable and verifiable science that allows human knowledge to expand. Isn’t this the goal after all?

Acknowledgements

The authors acknowledge the financial support from the Slovenian Research Agency for research core funding for the programme Knowledge Technologies (No. P2-0103).

References

https://doi.org/10.17226/25303.


Prinz et al. 2011, Believe it or not: how much can we rely on published data on potential drug targets?”. *Nature*, 2011. Available: https://www.nature.com/articles/nrd3439-c1 [July 2020].


Extending the Philosophy of Computational Criticism

Jesse Roberts
Electrical Engineering and Computer Science
Vanderbilt University
Nashville, TN 37235 USA
Jesse.Roberts@Vanderbilt.edu

Douglas H. Fisher
Electrical Engineering and Computer Science
Vanderbilt University
Nashville, TN 37235 USA
Douglas.H.Fisher@Vanderbilt.edu

Abstract
We consider two works in computational criticism, a little studied but important area for computational creativity (Stiny and Gips 1978) & (Fisher and Shin 2019). The analysis results in the development of a more general, novel, extended model of computational criticism. Along with the more general model, we also provide the first formal attempt to discuss and distinguish meaningful computational criticism. This theoretical synthesis should be useful to the study and development of computational critics and help to spark further dialogue in this area.

Introduction
What makes a person a critic? The United Nations Universal Declaration of Human Rights, Article 19 states “Everyone has the right to freedom of opinion and expression”. This is criticism at the most fundamental, possessing and expressing an opinion. Quite literally, everyone’s a critic; however, this does not mean that all opinions, or the critics which hold them, are to be considered equal.

In (Mendelsohn 2012), Daniel Mendelsohn, himself a world renowned critic, gives a simple formula for what makes a criticism meaningful, Knowledge + Taste = Meaningful Judgement. This makes intuitive sense. A child with no knowledge of painting or impressionism voicing their distaste for Monet should not carry the authority of a qualified art historian. The value assigned to the child’s criticism is less than that of the art historian because the child lacks knowledge. Therefore, while everyone is a critic not every criticism is meaningful.

With the above discussion in mind, this paper builds on two key works to attempt to formally distinguish meaningful criticism from the trivial. The focus herein may seem narrow with much of the ideas stemming from two papers. However, it is important to note that this paper is only the second paper in the history of ICCC to cite Algorithmic Aesthetics (Stiny and Gips 1978). The first was (Fisher and Shin 2019) which is the other paper we will cite heavily. We believe that these two papers make up the substantive literature on theoretical computational criticism, particularly within the larger literature on computational creativity.

• (Stiny and Gips 1978) established disambiguating formalism for the philosophy of computational critique. Their work sought to allow for the discussion of every possible critic with no prescription regarding the content of the criticism. So, in the existing model, every computational agent in the ecosystem is a critic with no consideration of the critic’s meaningfulness.

• (Fisher and Shin 2019) presented a detailed discussion of the desiderata for a computational critic. These desiderata are used as a basis to examine characteristics requisite in a meaningful critic. In (Fisher and Shin 2019) the authors recognize the need for and recommend further comparison with (Stiny and Gips 1978).

We extend Stiny and Gips’ philosophical formalism by interpreting and incorporating Fisher and Shin’s desired attributes for a computational critic. The result is a more general model which we believe allows for the discussion of all possible computational critics and novelly attempts to divide the critical space in Figure 1 into those which are meaningful and those which are trivial. We additionally rely on studies from the cognitive sciences as a general model of computational criticism must also account for human-like critics given the assumption that the mind is fundamentally mechanical.

Based on the ICCC call for short papers, this is in part a “Debate Sparks” paper as we believe that overt treatment of computational criticism “deserves more attention from the community”. Issues of computational “criticism” are implicit in generative systems (Fisher and Shin 2019), but overt, explicit treatment of computational criticism has remained nearly unaddressed. This paper is additionally a “Nuggets and Gems” paper, because Stiny and Gips (1978) is a gem, which has been cited and discussed early on in computing and design venues, but has been largely untouched by the ICCC community.

Importance of Formalism
In (Stiny and Gips 1978) the authors cite three reasons for the development of their algorithmic formalism. First, it provides a common framework in which to study criticism and design. Second, the very act of solidifying thoughts into an algorithmic formalism exposes the assumptions and details that may have otherwise been unclear. Their final justification is that an algorithm makes the theoretical become testable by implementing and executing the algorithm.
In our work on computational criticism, we want to study the class of critics which provide meaningful criticism for a given artifact. Therefore, our development of the present formalism is similarly motivated by the need for a common framework, awareness of the assumptions and details, and testability.

The Existing Formalism

The authors of (Stiny and Gips 1978) appeal to the general model of the process of thought introduced in (Craik 1952). In that model, thought is seen as made of three constituent processes; translation, reasoning, and re-translation which correspond to the receptor, analysis algorithm, and effector respectively in Figure 2. The receptor, \( R() \), takes in an artwork, \( \alpha \), along with contextual information, \( C_i \), and produces a description, \( \delta \). The aesthetic system contains any and all contextual information stored in memory, \( C_m \), used to inform the analysis algorithm. The analysis algorithm, \( A() \), uses this information to produce an interpretation and evaluation, \( \iota \) and \( \epsilon \), based on \( \delta \). Finally, the criticism, \( \chi \), is generated by the effector, \( E() \). The model introduced by Stiny and Gips is here referred to as the S&G model.

From a theoretical standpoint, \( R() \) and \( A() \) are mathematical/algorithmic transforms which produce outputs in grammars suitable for the description, interpretation, and evaluation of \( \alpha \). The description, interpretation, and evaluation are represented by \( \delta \), \( \iota \), and \( \epsilon \) respectively. Importantly, \( \delta \), \( \iota \), and \( \epsilon \) are latent and not directly represented in the final criticism. \( E() \) is a transform which takes these latent representations and produces the final, expressed criticism, \( \chi \). We refer generally to \( R() \), \( A() \), and \( E() \) as critical processes.

Stiny and Gips place no constraint on the description, \( \delta \), and interpretation, \( \iota \), or on the grammar to be used to express them. However, the evaluation, \( \epsilon \), is expected to take the form of a numeric aesthetic value.

Critiquing the Stiny and Gips Model

The S&G model is believed to be suitable for the description of all possible critics. As discussed in the introduction, Figure 1 shows that only a subset of human-like and not human-like critics are meaningful. We turn now to discuss the desiderata given by (Fisher and Shin 2019) and how the constraints placed by the desiderata on the S&G model result in a more rigorous idea of computational criticism and begin to give form to the idea of “meaningful”.

Understanding the Medium

It is desired that a critic understand the structural medium for a given artwork and a subset of it’s formal characteristics. Therefore, a meaningful description, must be an encoding of the artwork which preserves a subset of the medium specific formal characteristics. As an example, if \( \alpha \) is assigned to be a novel and the \( \delta \) resulting from \( R(\alpha|C_i) \) is a block of text in ascii format with no paragraph or chapter structure, then those medium specific formal characteristics (paragraphs and chapters) are lost.

Consider a critic with receptor, \( R() \), which produces a description, \( \delta \), for a given artwork, \( \alpha \), with formal and informal characteristics encoded. If \( \delta \) is insensitive to all formal characteristics for a given \( \alpha \), the critic is not meaningful. For a more complete discussion of model sensitivity analysis see (Kucherenko and Iooss 2014). For simplicity, a function, \( f \), can be said to be insensitive to input \( \theta_i \), if \( \forall \theta_i, \frac{\partial f(\theta_i)}{\partial \theta_i} = 0 \).

Just as a receptor which produces a description without encoding any formal characteristics would not be meaningful, a receptor which was unable to exclude any irrelevant information would also not be meaningful. An easy way to check this is to compare the length of a binary string required to encode the artwork and context, \( L(\alpha|C_i) \), information against the length of the binary string required to represent the description, \( \delta \). Therefore, a critic with receptor, \( R() \), which produces a description with binary string length, \( L(\delta) \geq L(\alpha|C_i) \) is not meaningful.

Hypothesize about the authorial intent & Attempt to socio-historically situate the artwork

A critic should be aware of a creator for the artwork and produce a hypothesis regarding the creator’s intent. A critic should also attempt to produce an interpretation which is informed by the socio-historic context of the artwork and its creator. These two desiderata have been grouped because they both impose sensitivity requirements on the critical processes. Both require that \( A() \), when presented with context information regarding the author and socio-historical setting, allow the context information to affect the output. To exemplify, consider a human critic presented with two identical renderings of the Mona Lisa one of which she is told was painted by Leonardo da Vinci in the 16th and the other a forgery made today. The paintings are identical in every visual fashion. Therefore, \( \alpha_{da Vinci} = \alpha_{forgery} \). However, the critic she delivers for each of the paintings will differ due to the contextual information.

To formalize this idea, context information is either encoded in \( C_m \) or provided to the receptor as \( C_i \). Therefore, a critic with receptor, \( R() \), and analysis algorithm, \( A() \), which produces interpretation, \( \iota \), which is not sensitive to \( C_i \) and \( C_m \) is not meaningful. It can be said even more generally that any critic with a critical process \( (R(),A(),E()) \) which is wholly insensitive to any of the available inputs is not meaningful.
Stiny & Gips’ Model of Computational Criticism

Figure 2: Diagram of Stiny and Gips’ description of a computational critic (Stiny and Gips 1978) which takes da Vinci’s Mona Lisa as input and gives Georgio Vasari’s analysis (Vasari 2007) as output.

**Predict the public response** A computational critic should produce a prediction regarding the public’s response for a given artwork. The S&G model likewise requires a critic to produce an aesthetic evaluation, \( \epsilon \). Therefore, we place no constraints on \( \epsilon \) beyond requiring it to exist.

Historically, the prediction of public response started as mathematical aesthetic evaluation with the major works surveyed briefly in (Greenfield 2005). It began with the work of (Birkhoff 1933) who introduced the formulation \( M = O/C \) where \( O \) is the perceived order and \( C \) is the complexity. Shannon’s information entropy became a common measure of complexity and was eventually incorporated into (Stiny and Gips 1978) which referred to the ratio \( \frac{M}{M} \) as unity. Stiny and Gips considered the ratio of length of interpretation to description, \( L(\epsilon)/L(\delta) \), to be a useful approximation of \( M \).

This field came to be known as computational aesthetics and stems from the first EG Workshop on Computational Aesthetics in Graphics was convened to address the need for computational aesthetic metrics in computer aided design. They defined computational aesthetics as, “the research of computational methods that can make applicable aesthetic decisions in a similar fashion as humans can” (Hoenig 2005). This definition is very similar to Fisher and Shin’s idea of public response prediction, reinforcing the equivalence of evaluation and the prediction of public response.

**Reason about the criticism** The highest bar set by Fisher and Shin is that a computational critic should attempt to justify the criticism. One possible formulation, \( \gamma \in \epsilon \), would

Extended Model of Computational Criticism

Figure 3: Diagram of the novel extended description of a computational critic which takes Monet’s Houses of Parliament as input and gives Arsène Alexandre’s analysis (Alexandre 1921) as output.
consider the justification, $\gamma$, to be a component of the interpretation. However, this makes the assumption that $\gamma$ is calculable based solely on $\delta$ and $C_m$. This assumption precludes the possibility that the critic may attempt to explain the interpretation without consideration of the process, $A()$.

Disconnecting the explanatory process from the originating process is not a new concept in the world of explainable artificial intelligence (Wick and Thompson 1992). The motivation for the separation lies in a psychological methodology for dealing with introspective evaluation. In (Ericsson and Simon 1980), the authors discuss a human tendency to provide an introspective narrative to account for behaviour and analysis which is entirely disconnected from the originating process. Based on this psychological observation, a general critic must allow for explanation which is distinct from the originating process to allow for modeling human-like criticism.

A more general formulation considers $\gamma$ to be a novel component to the general critic model and takes the form $[\epsilon, \delta, \gamma] = A(\delta, C_m)$. In this way, the justification, $\gamma$, may arise solely as a consequence of $\delta$ and $C_m$ and/or it may arise out of a transform applied to $\epsilon$ and/or $\delta$. This adjustment to the S&P model is shown in the extended model diagram in Figure 3.

**Aesthetic System or Knowledge Base**

In the S&G model, the aesthetic system contains the information necessary to choose an interpretive and evaluative method. (Fisher and Shin 2019) also suppose a similar structure which they refer to as the knowledge base. However, in their view the knowledge base is accessible to both the receptive and effective processes as well as the analytic process. This should be the case in general as background knowledge has been shown to affect how humans perceive and respond (van Meeuwen et al. 2014). Therefore, in Figure 3 $C_m$ is accessible to all critical processes.

### Table 1: Constraints for Meaningful Computational Critics

<table>
<thead>
<tr>
<th>#</th>
<th>Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A critic with receptor, $R()$, which produces a description, $\delta$, which is insensitive to all formal characteristics for a given artwork, $\alpha$, is not meaningful.</td>
</tr>
<tr>
<td>2</td>
<td>A critic with receptor, $R()$, which is unable to exclude any irrelevant information from the description, $\delta$, for a given artwork, $\alpha$, is not meaningful.</td>
</tr>
<tr>
<td>3</td>
<td>A critic with any critical process ($R(), A(), \text{ and } E()$) which is wholly insensitive to any of the available inputs is not meaningful.</td>
</tr>
</tbody>
</table>

**Conclusion**

The desiderata from (Fisher and Shin 2019) have been applied to the model from (Stiny and Gips 1978) resulting in a more general extended model and a set of constraints useful for distinguishing meaningful computational critics from the trivial. The adjustments and extensions to the model are shown in Figure 3. The constraints to distinguish meaningful computational critics from those which are trivial are shown in Table 1.

Even though some critics may not be considered meaningful based on the criteria here, a subset of the critical processes may be meaningful. An example would be a critic which did not possess a meaningful description, but did possess a meaningful evaluation. Systems of this nature are numerous in the literature.

The constraints in Table 1 are only a subset of the constraints which would be necessary to fully define a meaningful critic. However, this represents the first effort to rigorously discuss and distinguish meaningful computational criticism and will hopefully prove useful in their future development.

**References**


On the Inherent Creativity of Self-Adaptive Systems

Simo Linkola, Niko Mäkitalo and Tomi Männistö
Department of Computer Science
University of Helsinki
Helsinki, Finland
{simo.linkola, niko.makitalo, tomi.mannisto}@helsinki.fi

Abstract

We argue that frameworks employed in architecting self-adaptive systems allow the system to exhibit creative behaviour, and that many of the existing self-adaptive systems operating in domains which are typically not associated with creativity are inherently creative. However, even the current state-of-the-art solutions do not fully exploit stronger forms of creative behaviour, which are required in complex environments, where the system constantly encounters fundamentally novel situations. To this end, software development necessitates a paradigm shift parallel to moving from procedural design methodology toward self-aware systems where the system adapts to its context at run time.

Introduction

Self-adaptive systems are software systems where a base system is monitored and adapted during runtime by another component, called a manager, so that the base system’s operation is maintained towards its goals in changing situations (see e.g. Salehie and Tahvildari, 2009). The self-adaptive systems have been studied extensively in the software engineering and architectures (see e.g. Kephart and Chess, 2003; Garlan et al., 2004; Kounev et al., 2017).

We interconnect the existing work on self-adaptive systems and computational creativity (cf. Colton and Wiggins (2012)). Particularly, we consider a host of models where a system may explore new and/or assess old (and new) adaptations at run time. We separate different types of creative adaptations and argue that a self-adaptive system must be self-aware (Kounev et al., 2017) for stronger forms of creativity. For implementing creative self-adaptivity, we illustrate an example control flow which extends MAPE-K loop (Kephart and Chess, 2003). The early results using an example self-adaptive system support the usefulness of explicitly considering the self-adaptive system’s creativity.

Creativity in Self-Adaptive Systems

Assessing a system’s creativity depends on the adopted definition of creativity (Jordanous, 2012) and the assessment perspective (Jordanous, 2016). We adopt the standard definition: creativity is the ability to produce novel and valuable ideas or artefacts (Boden, 1992; Runco and Jaeger, 2012), in our case adaptations\(^1\). We extend the definition with the notion of intentionality (Ventura, 2017) as the creative process of a system should not be fully random (cf. creative autonomy (Jennings, 2010)).

Definition. Creative self-adaptivity is the ability of the system to intentionally adapt in ways which are valuable and novel.

To this end, the designer needs to provide the self-adaptive system means to appear creative by relaxing the soft constraints on the adaptive behaviour, increasing flexibility. Critically, the relaxed adaptive freedom should not result in erratic behaviour. The system should seek creative adaptations only when it reasons that it does not know of an acceptable solution to the current (or expected future) situation.

Prominently, to support creative behaviour in complex environments, we need to engineer on-going learning capabilities to the system, i.e. the system needs to be self-aware (Agarwal et al., 2009; Kounev et al., 2017).

In this paper, novelty, value and intentionality are inspected and argued through the system’s internal reasoning processes. However, as the system’s behaviour should reflect its higher-level design goals, our presumption is that strengthening system’s own understanding of its creative and adaptive process will reflect in its designer granting more creativity to it.

Intentionality of the system’s adaptation process is typically well justified in a self-adaptive system as the system adapts only when it perceives a clear reason to do so. However, if the system adapts only reactively, the situation may change before the adaptation completes, ultimately requiring different adaptation strategy (cf. Moreno et al. (2015)). For stronger intentionality, the system needs to be able to adapt proactively and verify candidate adaptations at run time.

Value of an adaptation corresponds to the evaluation

\(^1\)By ‘adaptation’ we refer to all of the following: a configuration of the (base) system, behaviour the configuration causes in a situation, and the act of deploying the configuration in a particular situation. Ambiguities are pointed out where appropriate.
of the system’s behaviour in a particular situation, e.g., with respect to reliability and performance (Muccini et al., 2016). Naturally, value depends on the system’s operational domain and it translates to, e.g., adhering to the hard configuration constraints of the base system (e.g., by using constraint evaluator as in Rainbow architecture (Garlan et al., 2004)), and/or adapting in a way which maximises the expected utility of the system’s behaviour to some finite time horizon (e.g. Moreno et al., 2015).

Novelty of an adaptation is not commonly scrutinised in self-adaptive systems. The need for novel adaptations rises from novel situations the system encounters. The system may also explore new adaptations (or try existing ones in new situations) when it has free resources, e.g., using simulations which capture relevant aspects of the world w.r.t. its operational goals.

We separate three conceptually different senses of how an adaptation may be novel: novel strategy, novel configuration and novel transformation. Figure 1 shows an example of how a MAPE-K loop (Kephart and Chess, 2003) can be expanded to include separate processes for all novelty types in a sequence. In an actual system, however, they may be mixed within the same generic process or run in parallel.

Novel strategy means that an existing adaptation is deployed in a novel situation, which resembles combinational creativity (Boden, 1992). Novel strategies are pervasive in existing systems, as the systems are commonly engineered to analyse any encountered situation and select the most fitting adaptation for it (Salehie and Tahvildari, 2009).

Novel configuration is generated by a search-based method during the system’s execution, which is analogous to exploratory creativity (Boden, 1992). The system may aim to find novel and valuable configurations for particular situations or generally well performing ones. Existing systems utilise such methods, e.g., by combining composite adaptations from elementary adaptation tactics (Moreno et al., 2015) or using AI search methods to find novel adaptations (Hadaytullah et al., 2012).

Novel transformation alters the whole system in a new way, requiring same kind of mechanisms as transformational creativity (Boden, 1992; Linkola et al., 2017). In general, the transformation can be adjusting goals or any other part of the system, hence, flexible system validation and verification at run-time becomes paramount. For apt transformations, the system needs to have significant and timely understanding of its own operation and its relation to the environment, e.g., through self-modelling (Kounev et al., 2017, Chapter 9) and simulated behaviour.

Self-awareness (Kounev et al., 2017; Linkola et al., 2017) is a major enabler of creative behaviour, aiding the system to reason about value, intentionality and novelty, and act based on its reasoning. It is essential especially for novelty for inferring how and where two adaptations differ from each other, by their configuration and their behaviour (in particular situations), allowing to estimate how configuration changes affect the behaviour. Ultimately, new situations can be compared to previously encountered ones, and new configurations and transformations can be compared to previously deployed ones with an understanding of how reliable the comparison is. For example, when a robot is deployed in a new location, the system can compare its previous locations to the new one and rapidly orient itself to the new environment by selectively adapting parts of its configuration.

Particularly, transformational creativity becomes a requirement when shifting self-adaptive systems towards general AI paradigm as the system needs to be able to change its goals and be flexible in its reasoning based on its own experiences. To this end, we need rigorous software engineering practices to guide the development, e.g., to ensure safer (and ethical) behaviour (Winfield, 2014). Next, we distinguish some aspects of self-awareness which have direct influence on creative behaviour.

Self-Awareness Components

In computer systems, self-awareness can be considered as an umbrella term, enclosing a variety of different but partially overlapping aspects, many described in detail in Kounev et al. (2017). In computational creativity, these aspects have been considered with respect to metacreative systems (Linkola et al., 2017). We include three distinct awarenesses, and explicate how they contribute to the goals of creative self-adaptation: novelty, value, and intentionality.

Goal-awareness ensures that the goals are coherent after their alteration to support intentional and valuable adaptations. The task is challenging since no component may have a direct knowledge of every goal. For this reason, the system can maintain a goal model which has an immediate influence on the evaluation function guiding the adaptation search. Thus, the goal changes reflected in the evaluation function bias the subsystems’ behaviour towards adaptations conforming to the modified goals. Understanding the novelty of new goals allows the system to find the most similar previous goals

---

Figure 1: Example search flow for novel and valuable adaptations realised within the manager component.
and use their deployed adaptations from its memory as a starting point for, e.g., novel configuration search (see Figure 1).

**Context-awareness** ensures that the system has the ability to understand the novelty of the system’s current context to support intentional adaptations. For this purpose, context-awareness uses the input data provided by the base system, which then influences evaluation function. It can also motivate adaptations by enabling the base system to switch among external services with different behavioural or quality profiles. Context-awareness can also directly influence novel adaptation search, supporting context relevance and reducing search complexity.

**Resource-awareness** maintains the base system’s current input and output mechanisms (e.g., sensors, actuators) as well as the data and control connection among the resources. Additionally, for making an adaptation intentional, a resource-centric sub-evaluation can be used to further filter out the adapted systems with excessive resource demands.

**Example: A Light Control System**

We use a community greenhouse light control system as an example Base System to demonstrate the usefulness of creative self-adaptation, focusing on how deliberate reasoning of the novelty (of the context, etc.) enhances the self-adaptive system’s operation.

A community greenhouse is a shared space where people may rent patches to grow their own plants. Each patch is square shaped with its own lamp. For the sake of the example, we assume that plants in each individual patch are approximately of the same height, but the height varies from patch to patch. The goal of the system is to produce certain amount of light (lux) to the top of the plants in each patch. The system can adapt to different situations (ambient light level and plant height per patch) by turning lamps on and off.

The system has a set of base adaptations which are given by the system’s designer, and it searches for novel configurations (lamp settings) in novel situations using the ($\mu + \lambda$) evolutionary algorithm (Back et al., 1997). When the system deploys a novel configuration, it stores it to its memory with information about its deployment situation. Awarenesses are used to gauge which of the already deployed adaptations are used to induce the search in a new situation and how the adaptations are evaluated.

We consider the following systems, with their example adaptations shown in Figure 2:

**S1:** A goal-aware system perceives the ambient light level across the greenhouse and knows the target light level. The system assumes that the plant tops are always on the same level and that the height does not alternate from patch to patch.

**S2:** A goal- and context-aware system knows the target light level and uses sensors to measure the height of the plants in each patch.

**S3:** A goal- and context- and resource-aware system, in addition to S2, aims to minimise the number of lamps that are on.

We run each system 10 times through 2000 situations with varying plant heights and ambient light levels, reporting the averages for the first 1000 (fh) and the last 1000 (lh) situations in Table 1. The implemented awarenesses benefit the system by keeping it closer to its target light level (Lux Avg, marked as difference from the target) with decreased variance in the light level (Lux Std)).

In addition, S2 and S3 are able to efficiently utilise the previous contexts. Lux Avg of S2 approaches the target light level when comparing the last half to the first half, and both S2 and S3 are able to decrease Lux Std. Thus, the later adaptations of S2 and S3 are more apt because of the learned models, whereas S1 struggles to learn anything valuable from the ambient light level alone. Further, S3 is able to keep closer to the system’s goal (Lux Avg, Lux Std) than S2 or S1, and does it by using less electricity (Lamps), saving both resources and money. Notably, the time used to adapt (Time, in seconds) does not increase with the added complexity of the implemented awarenesses (S2, S3). The detailed novelty assessment enables the system to filter out previous adaptations based on their perceived distance to the current situation, decreasing time used to process adaptation’s in memory.
Table 1: Each system’s statistics.

<table>
<thead>
<tr>
<th>System</th>
<th>Lux</th>
<th>Avg Lux</th>
<th>Lux Std</th>
<th>Lamps</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1 (fh)</td>
<td>+494</td>
<td>2701</td>
<td>191</td>
<td>0.486</td>
<td></td>
</tr>
<tr>
<td>S1 (lh)</td>
<td>+499</td>
<td>2704</td>
<td>192</td>
<td>0.506</td>
<td></td>
</tr>
<tr>
<td>S2 (fh)</td>
<td>+405</td>
<td>2708</td>
<td>192</td>
<td>0.468</td>
<td></td>
</tr>
<tr>
<td>S2 (lh)</td>
<td>+309</td>
<td>2610</td>
<td>192</td>
<td>0.475</td>
<td></td>
</tr>
<tr>
<td>S3 (fh)</td>
<td>+29</td>
<td>2616</td>
<td>181</td>
<td>0.491</td>
<td></td>
</tr>
<tr>
<td>S3 (lh)</td>
<td>-27</td>
<td>2530</td>
<td>182</td>
<td>0.492</td>
<td></td>
</tr>
</tbody>
</table>

Conclusions

We have argued that creativity is inherent in self-adaptive systems. Using an example system, we have shown that even simple mechanisms for novelty assessment can help, e.g., to improve value and decrease resource usage. However, especially for transformational novelty, we need to change our approach to software development, beginning from specifying the system requirements to allow more pronounced creative behaviour. This, in turn, places substantial demands on the system verification and validation at run time.

Acknowledgments

This work has been funded by Academy of Finland grants 313973 (CACS) and 328729 (CACDAR). We thank Hadaytullah Kundi, Ph.D. for his contributions towards this work.

References


Towards Enhanced Creativity in Interface Design through Automated Usability Evaluation

Snehal Dhengre¹, Jayant Mathur², Farzaneh Oghazian³, Xiaomei Tan¹, Christopher McComb²

¹Department of Industrial and Manufacturing Engineering, ²School of Engineering Design, Technology, and Professional Programs, ³Department of Architecture
The Pennsylvania State University
University Park, PA 16802 USA
sud702@psu.edu, jmm8886@psu.edu, fxo45@psu.edu, xzt24@psu.edu, mccomb@psu.edu

Abstract
With an increase in the number of mobile apps making their way to users, there is a growing need for tools to support the app design process. While many tools focus on increasing the pace of development, few attempt to aid the designer in generating more creative solutions. In this work, we take creativity as the combination of novelty and utility. Particularly during development of user interfaces, assessment of utility (primarily usability) is iterative, rigorous, and time-consuming. The objective of the proposed work is to explore and evaluate the use of machine learning to predict usability measures for mobile app interfaces as a means to automate usability evaluations. Specifically, a convolutional neural network (CNN) is used to accurately (nearly 90%) predict three usability measures: regularity, complexity, and touchability. This tool automates the assessment of utility in app design, freeing up the designer to seek designs that are novel and thus creative.

Introduction
Consumers downloaded 204 billion mobile applications (apps) worldwide in 2019 (Clement 2020) and the usage of mobile apps is continuing to increase at a tremendous rate of over 10 apps used per day on average by a user (Annie 2017). Mobile app users are therefore becoming selective due to the abundance of mobile apps, many of which contain similar features and services. In order to keep up with the consumer demand for engaging mobile apps, designers must create interfaces that are creative - both novel and usable. However, utility testing during user interface (UI) development limits the time and resources designers can allocate toward novelty. This utility testing is typically done through usability evaluation. The usability of an interface is defined as “the extent to which a product can be used by specified users to achieve designated goals with effectiveness, efficiency, and satisfaction in a specified context of use” (ISO9241-11 2018).

Current methodologies to evaluate usability often entail empirical assessments of observations, but these evaluations can be costly, time-consuming, and resource intensive. These limitations hinder the creativity of the designers as significant cognitive effort must be expended to ensure utility instead of pursuing novelty. There is, therefore, an opportunity to introduce automation in usability evaluation during UI development to help designers become more creative by empowering them to spend more time and effort pursuing novelty than ensuring utility. This work therefore aims to aid in improving design creativity by helping designers expedite the quantitative evaluation step in the design cycle, thereby providing designers more time and resources to focus on novelty. The goal for the current work is to investigate the efficacy of an automated evaluation tool to predict usability metrics, a step towards developing a proof of concept that will enhance creativity in interface design.

Related Work
Research in the utilization of computational tools for automation, intelligent feedback generation, and assistive design guidance shows that designers may benefit creatively when relying on computational assistive tools (Colton et al. 2019; Colton, Powley, and Cook 2018; Karimi et al. 2019). Colton et al.(2019; 2018) explain that computational automation can “empower people in a co-creative setting” thereby allowing designers to think outside their perspective and aiding creativity. Such co-creative design tools driven by machine learning (ML) can provide innovative solutions to enhance and aid in the development of creativity and creative designs (Karimi et al. 2019; Verganti, Vendraminelli, and Iansiti 2020). Moreover, ML is gaining momentum in the field of human-computer interaction (Yang, Banovic, and Zimmerman 2018) and offers potential for innovation in user-centered design as well. Various research studies have been conducted to enhance the process of usability evaluation. For instance, PLAIN, a java-based automatic evaluation plugin, was developed for calculating the quality of the interface from a usability perspective (Soui et al. 2017). Using a similar approach, Soui et al. (2019) developed a multi-objective automatic optimization method to detect aesthetic aspects of UI. In their study, two sets of mathematical measurements for usability, guidance and coherence, were introduced as a tool for qualitative assessment of graphical user interface (GUI) (Soui et al. 2019).

While some studies incorporated quantitative methods for usability assessment of UIs and validated the methods through empirical studies, others implemented data-driven strategies to identify mobile app issues and automate the process. A data-driven based tool, ZIPT, was developed...
to measure the performance of mobile app interfaces using metrics such as completion rate, time on task, and the number of interactions performed (Deka et al. 2017b). TapShoe was developed by Swearngin and Li (2019) to predict the tappability of UI elements. Both of the aforementioned studies used crowd-sourcing for collecting data and ML models for assessing usability.

Aesthetics are strongly associated with the functionality and user satisfaction of an interface (Kurosu and Kashimura 1995; Tractinsky 1997; Hartmann, Sutcliffe, and De Angeli 2008). Norman (2002) explains that good aesthetic design has a positive impact on user experience, thereby enhancing usability of the interface. These benefits of good aesthetic design have led researchers to explore the aesthetic measures of an interface to assess usability. For instance, Ngo, Teo, and Byrne (2000) defined 14 aesthetic measures that closely aligned with designers’ implicit perceptions of aesthetics (Ngo, Samsudin, and Abdullah 2000).

Accessibility for interactive elements is another important usability attribute, particularly with touch screen devices where touch target size influences user’s accessibility of the interface. Smaller touch targets require a higher time to tap and often leads to frustration. Parhi, Karlson, and Bederson (2006) recommended that a minimum size of 1 cm x 1 cm is required for accurate and efficient selection of touch targets. Similarly, Microsoft recommends a target size greater than 0.9 cm x 0.9 cm for frequently used interactive elements (Microsoft 2012).

### Data Extraction and Augmentation

This study uses a subset of Rico dataset (Deka et al. 2017a). The dataset contains visual, textual, structural, and interactive design properties of 72,000 screens from mobile apps. Specifically, this study was conducted with annotated images of weather apps with 5 or 6 UI components (205 screens). Data was augmented by adding gaussian blur to the extracted images to increase the diversity of the dataset. This resulted in a total of 410 annotated RGB images of size 1440 x 2560 pixels for weather apps category.

Three measures (regularity, complexity, and touchability) were used as evaluation criteria to assess the usability of the mobile apps. These measures were selected to highlight the opportunity for using ML as an automated usability evaluation tool and do not resemble a holistic usability evaluation approach. Regularity measures the consistency of organization of the UI components and spacing between UI components which is given by:

\[
RM = 1 - \left( \frac{N_{av} + N_{ah} + N_{sp}}{3n} \right) \in [0, 1] \tag{1}
\]

where \(N_{av}\) and \(N_{ah}\) are the number of horizontal and vertical alignment points, \(N_{sp}\) the number of distinct distances between column and row starting points, and \(n\) the number of components (Soui et al. 2017).

Complexity measure determines how easily a user can find expected information and is defined in terms of the number of components and number of alignment points on a UI. The complexity measure is given by (Soui et al. 2017):

\[
CM = \frac{N_{av} + N_{ah}}{(2n)} \in [0, 1] \tag{2}
\]

Touchability measures the efficiency and accuracy of interaction with touch targets with regard to accessibility of UI and is defined by the authors as:

\[
TM = \frac{N_c}{N_{TOT}} \in [0, 1] \tag{3}
\]

where \(N_c\) is number of clickable components meeting the minimum touch target size requirement, and \(N_{TOT}\) the total number of clickable components.

To calculate these usability measures, the component class of each UI component (clickable or non-clickable), and the position of starting point and endpoint of each UI element were extracted for each mobile app image. Using these values, new index parameters were computed for each mobile app image including the number of vertical and horizontal alignment points, the number of distinct distances between row and starting points, the number of UI components, and the number of clickable components. An example of these index parameters for an image is demonstrated in Fig. 1. The usability measures, RM, CM, and TM, were then calculated using mathematical Eq. (1), Eq. (2) and Eq. (3), respectively. The minimum size for the touch object used for calculation was 0.9 cm x 0.9 cm.

![Image](image.png)

**Figure 1:** An example of index parameters for a mobile app page

### Design and Implementation

Convolutional neural networks, like the model used in this paper (see Fig. 2), are a class of deep learning networks proven to be very effective for image recognition. Recently, such models have seen increased use in design (Williams et
The specific model in this work was trained on 70% of the images from the generated dataset and tested with the remaining 30% of the images. Two layers of convolution and max pooling were applied on each mobile app screen, following which the model was forked into three branches to predict the three usability measures. Another layer of convolution and max pooling was applied for the branch predicting touchability measure for obtaining finer resolution of the input image. Sigmoidal and hyperbolic tangent activation functions were used for the output layer of RM and CM, and TM respectively, to limit the output bounds between 0 and 1. The respective activation functions ensured best fit for the model. A mean squared error loss function was used along with the Adam optimizer to compile the model.

To evaluate the performance of the model, the R-squared score was used as the performance metric, which assesses how close the predicted values are to the actual ones. The model was evaluated for the training and testing data separately by computing a coefficient of determination (R-squared score) for each of the three usability measures.

**Results**

The R-squared score computed from the best trained model was greater than 90% for the training set and greater than 85% for the test set for all three usability measures. The corresponding R-squared score for each usability measure is shown in Table 1. These results suggest that the machine learning approach allows designers to predict close to accurate predictions to the expected usability measurement values. Prediction results of a sample mobile app screenshot is shown in Fig. 3, where low RM corresponds to fewer alignment points, high CM corresponds to difficulty in finding the required information, and low TM corresponds to fewer number of touch targets meeting minimum size criteria.

<table>
<thead>
<tr>
<th>Measure</th>
<th>R-squared score (%)</th>
<th>Train set</th>
<th>Test set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regularity</td>
<td>92.68</td>
<td>88.57</td>
<td></td>
</tr>
<tr>
<td>Complexity</td>
<td>92.17</td>
<td>89.11</td>
<td></td>
</tr>
<tr>
<td>Touchability</td>
<td>98.30</td>
<td>89.18</td>
<td></td>
</tr>
</tbody>
</table>

**Conclusion**

The goal for this work was to free up designers from spending time and resources on utility testing of mobile app UI designs and focus more on the novelty of designs, thereby improving the collective creative output. The proposed work therefore investigated automation of the usability evaluation of mobile app UIs. A framework for predicting usability measures for mobile apps using ML was presented as a tool for designers to conduct rapid usability evaluation. Specifically, a convolutional neural network model was developed to predict three usability measures: regularity, complexity, and touchability when given an input image of the mobile app UI. The findings indicate that ML can be used as a means of automation in the development of tools that assess the usability of mobile app UIs. There is potential to integrate such tools with GUI design platforms such as Adobe XD, Grasshopper, and Balsamiq. This can support designers by providing a quick usability assessment of interface designs, thereby allowing designers to instead focus their efforts on the novelty aspect of their designs.

The findings in the present study should be considered in light of some limitations such as small size of dataset used for training the ML model. Future work could explore validation of the tool by conducting usability study to assess its efficacy. Also, addition of mobile app screens from other platforms and applications would provide a broader basis for generalizing the findings.

---

app categories and incorporation of other usability measures could be considered to build a robust usability evaluation tool.

References


Microsoft. 2012. UI design and interaction guide for windows phone 7 series.


Yang, Q.; Banovic, N.; and Zimmerman, J. 2018. Mapping machine learning advances from hci research to reveal starting places for design innovation. In CHI’18.
From Computational Creativity to Creative Problem Solving Agents

Evana Gizzi\(^*\),\(^1\) Lakshmi Nair\(^*\),\(^2\) Jivko Sinapov\(^1\), Sonia Chernova\(^2\)

\(^1\) Department of Computer Science, Tufts University, MA, USA
\(^2\) College of Computing, Georgia Institute of Technology, GA, USA
Evana.Gizzi@tufts.edu, lnair3@gatech.edu, Jivko.Sinapov@tufts.edu, chernova@cc.gatech.edu

Abstract
Creative problem solving (CPS) is a skill that can greatly improve resourcefulness and adaptability of existing artificial intelligence (AI) systems. In this paper, we discuss how CPS leverages theoretical aspects from Computational Creativity (CC) and planning in AI. We present a definition of CPS, and discuss how CPS is achieved using aspects of CC and problem solving in AI.

Introduction
The Apollo 13 incident of 1970 is an example of how human ingenuity and creativity saved the lives of the three astronauts on-board. In order to combat the increasing carbon dioxide levels in the spacecraft, the astronauts crafted a carbon dioxide filter using available objects (Cass 2005). Similar capabilities are currently beyond the scope of existing artificial agents. In this paper, we focus on Creative Problem Solving (CPS) - a skill that can greatly improve resourcefulness of existing artificial intelligence (AI) systems. We discuss how CPS adapts theoretical aspects from Computational Creativity (CC) and general problem solving in AI, thus combining the two. We focus specifically on agents that plan and learn over states and actions, in the context of creative problem solving. We adapt terminologies frequently used in the planning and learning literature in AI, and link them to theoretical aspects in CC. While there have been existing efforts at formalizing CPS, the formalizations have focused specifically on either AI (such as classical planning (Sarathy 2018; Erdogan and Stilman 2013)), or perspectives in CC (such as concept re-representation (Olteteanu 2015; 2014)). In contrast, we present a formalization of CPS by adapting aspects of problem solving from both AI and CC. This interdisciplinary approach allows us to take a holistic perspective on CPS, to stimulate further research in the area.

Definition of Creative Problem Solving
We begin by defining the components of a problem to be solved by an agent acting in its environment through planning or learning. The planning or learning problem specification in AI typically consists of a task goal \( G \) to be accomplished, given a set of environment states \( S \) and agent actions \( A \). The

\( ^* \)Both authors contributed equally
initial knowledge is limited. Note that $C^X = C^X^*$ is often not a practical assumption for real-world agents, since it implies that the agent knows every concept possible.

A crucial aspect of CPS that differentiates it from general problem solving is that the initial conceptual space $C^X$ known to the agent is insufficient to accomplish the task goal. Traditional planning or learning approaches in AI often yield a failure in these circumstances. For example, the agent might need to learn about a completely new state or action to be able to accomplish the task. Thus, CPS is characterized by its flexibility or adaptability to handle novel problems (Guiford 1967). We formally define CPS as (see figure 1):

**Definition 1** Creative problem solving is defined as the process by which the agent discovers new concepts that were not in the initial conceptual space of the agent, allowing it to accomplish a previously impossible goal. Formally, CPS refers to the process by which the agent discovers new concepts $c^X \in C^X$, where $C^X \not\subseteq C^X^*$, such that $C^X$ enables the agent to solve a task that was previously unsolvable when only $C^X = C^X^*$ was known.

In other words, the space of concepts that is explicitly represented by the agent defines the boundaries of what the agent can plan to accomplish. Creativity arises when the agent uses what it already knows, to discover something new. In the context of problem solving, the newly discovered knowledge is applied to solve a previously impossible task. In the following sections, we expand upon our definition, highlighting the theoretical aspects in CC that apply to CPS.

**Adaptations of Aspects in Computational Creativity to Creative Problem Solving**

Although there is no widely accepted singular definition for computational creativity (Jordano 2012), there do exist generally accepted aspects. In this section, we review four major aspects of CC, and their inheritance and adaptations to creative problem solving. The aspects listed below are grouped into two categories: output-based aspects and process-based aspects. These categorizations are not meant to divide types of systems, but rather, to group key aspects.

**Output-based Aspects**

In output-based aspects, the focus is on evaluating the creativity of a system by determining whether the output produced in a task is considered to be creative. These systematic outputs, referred to as artefacts in the CC community, may take physical and/or non-physical form (e.g. paintings, songs, recipes). The first aspect (Novelty and Value) describes two key characteristics of a creative output, whereas the second aspect (Evaluative Methods) describes methods for making the evaluations.

**Novelty and Value:** This first aspect stems from the work of Margaret Boden, who proposed that creativity necessitates both novelty and value (Boden 1998). Novelty guarantees that the generated outputs of a creative process are in some way original, whereas the value criteria ensures that the generated outputs are not just random generations, but in some way geared towards a creative goal. Both novelty and value have contextual considerations. For example, an agent may produce a novel painting, but in the context of a scenario which calls for a creative recipe, the novel painting would not be considered valuable (Sosa and Gero 2016; Varshney, Wang, and Varshney 2016).

**Evaluative Methods:** In evaluative methods, the creativity of a system is evaluated by judging the output of its processes, deeming them to be creative or not creative. Similiar in nature to the Turing test, these methods focus on using the judgement of an observer on the product of a creative process. The evaluation can either happen computationally (Colin et al. 2016; Colton, Wiggins, and others 2012; Varshney, Wang, and Varshney 2016), from a human evaluator (Bishop and Boden 2010; Guckelsberger, Salge, and Colton 2017), or from a social group (Varshney, Wang, and Varshney 2016). The nature of these evaluations vary, where in some cases the output is compared to a human’s creative output, and in others, the output is judged in a social context. Additionally, the evaluation of a creative output can be determined by the agent’s ability to explain its own intentions and motivations to a human (Cook et al. 2019).

**CPS Adaptation of Output-based Aspects:** In the context of problem solving, the novelty criteria is fully inherited. Creative solutions may not be completely original themselves, but rather in their application to the problem. For example, using Tupperware as a container may not be original in itself, but using Tupperware as a replacement for a soap dish may be considered a creative solution to a problem. Formally, a concept $c^X$ is said to be novel when it is not contained within the initial conceptual space $C^X$ of the agent for that problem, i.e., $c^X \not\subseteq C^X$. The second criteria in CC is that creativity necessitates value. In the context of CPS, this criteria is inherited as usefulness or utility. That is, does the solution actually solve the problem? A conceptual space $C_X^*$ is said to be useful, when the goal state $G$ can be accomplished via concept(s) $c^X \in C_X^*$.

CPS does not directly inherit evaluative methods. This is because the output of a CPS process is simply evaluated as either successful or not successful, based on its ability to solve the problem. As such, a successful solution to a problem which necessitates CPS is inherently creative. Thus, evaluation in CPS involves evaluating whether the new conceptual space $C_X^*$ is sufficient to accomplish the current goal.

**Process-based Aspects**

Process-based aspects focus on the method by which creative output is generated. Contrary to output-based, process-based aspects are concerned with the question of how outputs are produced as opposed to the evaluation of what is produced. The first aspect (Procedural Methods) reviews existing methods for synthesizing the creative process, whereas the second aspect (Boden’s Types of Creativity) reviews three ways of implementing procedural methods.

**Procedural Methods:** In procedural methods, the focus lies on evaluating creativity based on the method by which
it systematically generates its creative output. These methods have been explored across many processes, ranging from machine learning approaches (Toivonen and Gross 2015), to associative algorithms (Varshney, Wang, and Varshney 2016), and autonomous evaluation algorithms (Jennings 2010). A popular approach in CC is a two part method, consisting of an expansion phase where the agent synthesizes a large set of possible outputs for a creative process, and a contraction phase where the agent processes the candidate outputs in order to select valuable output. Analogous conceptualizations of the expansion phase include divergent thinking, generative thinking, and defocused attention. Analogous conceptualizations of the contraction phase include convergent thinking, evaluative thinking, and focused attention (Guilford 1967; Pereira and Cardoso 2002; Zhang, Sjoerds, and Hommel 2020; Sarathy 2018).

**Boden’s Types of Creativity:** Boden proposed three categories that describe the process of generating creative outputs, namely, combinational creativity, transformational creativity, and exploratory creativity (Boden 1998). Combinational creativity involves taking known or familiar information, and combining it in a way that generates a novel output (Pereira and Cardoso 2002; Lieto et al. 2019). Transformational creativity involves transforming one or more dimensions of the solution/output space to provide the means for new structures to emerge in the transformed space. Lastly, exploratory creativity involves an exhaustive search of a solution/output space to find a novel solution.

**CPS Adaptation of Process-based Aspects:** Creative problem solving directly utilizes process-based approaches. CPS is typically triggered by an impasse moment, where the agent detects that nominal problem solving techniques are insufficient for accomplishing the goal (Knoblich et al. 1999). Impasse is followed by a period of incubation, where the agent generates the solution space, synthesizing possible ways of solving the problem using a relaxed representation of the problem and domain. Once a viable solution is found in this space, the agent is said to reach its insight or “Ahah!” moment (Colin and Belpaeme 2019), wherein the agent proceeds to use the solution to solve the problem. We call this process the impasse-incubation-insight process. While there exist other general formalizations of the creative process (Mumford et al. 1991; 1997), we use the impasse-incubation-insight paradigm to facilitate our adaptations. The impasse-incubation-insight process can be implemented using the two part method of expansion and contraction in the following manner – the impasse moment triggers incubation, where the agent enters the expansion phase and generates a new conceptual space \( C'X \) from \( CX \). This is followed by the contraction phase, where the agent applies the newly discovered concepts \( C'X \) to generate a plan for accomplishing the goal (insight moment).

Boden’s types of creativity are inherited into creative problem solving by providing three ways to expand an agent’s initial conceptual space. Thus, Boden’s types of creativity can be applied to manipulate the initial conceptual space \( CX \) of the agent, in order to come up with a new conceptual space \( C'X \) for solving the problem.

**Combinational methods** involve combining existing information in an agent’s conceptual space to generate a novel conceptual space for solving the problem. The agent creates new concepts \( c'X \in C'X \) by combining existing concepts in \( CX \). We define a function \( f \) that combines concepts in \( CX \) to create the new conceptual space, such that \( f(c'X) = c_iX \), when more than one concept is not combined:

\[
f : CX \rightarrow C'X \mid f(X) = f(c_1X, ... c_kX);
\]

\[
c_1X, ..., c_kX \in CX, C'X \nsubseteq C'X
\]

If \( c_1X, ..., c_kX \nsubseteq C'X \), we can redefine a new conceptual space of \( \bigcup_{i=1}^k c_iX \), where combinational creativity applies. In CPS, combinational creativity can be observed when learning new behaviors or skills as a composition of previously known behaviors (Hang et al. 2020), or constructing new tools by combining objects (Nair, Balloch, and Chernova 2019).

**Transformational methods** involve transforming the problem representation in some way to generate a novel and previously unknown representation of the same problem, i.e., the agent transforms the initial conceptual space \( CX \) into a new conceptual space \( C'X \). The set of concepts \( c'X \in C'X \) can be represented as follows:

\[
f : CX \rightarrow C'X \mid c'X = f(cX) \forall cX \in CX, C'X \nsubseteq C'X
\]

Thus, \( f \) denotes a surjective function that maps every concept in \( cX \in CX \) to a new concept \( c'X \in C'X \). Transformational creativity involves mapping from the initial conceptual space to a new conceptual space, via an appropriate transform, e.g., rotations or translations (Fitzgerald, Goel, and Thomaz 2017), and segmentations (Gizzi, Castro, and Sinapov 2019).

**Exploratory methods** involve searching the universal conceptual space \( C'X \), to discover a novel solution. The agent may discover a new conceptual space \( C'X \subseteq C'X \) either via random exploration of its environment (i.e., babbling), or guided exploration using heuristics or loss or reward functions. If the agent uses a loss function, the concepts \( cX \in C'X \) can be represented as follows:

\[
\{cX = \arg\min_{cX} \mathcal{L}(cX) \text{ s.t. } cX \in C'X \}
\]

where \( \mathcal{L} \) denotes an appropriate loss function, and \( C'X \) contains novel concepts from the universal conceptual space such that \( C'X \nsubseteq C'X \). In general, approaches that explore the state space to derive a solution, e.g., reinforcement learning (with reward functions), search through planning spaces (Erdogan and Stilman 2013), and motor babbling (Sinapov and Stoytchev 2007), fall into this type. In large conceptual spaces, this form of creativity can be prohibitive.

**Conclusion**

In this paper, we presented a formal definition of creative problem solving as the intersection of computational creativity and problem solving in AI. We specified key aspects of CC systems, and formalized their adapted inheritance into CPS. Research in creative problem solving has taken place mostly within the confines of the artificial intelligence community, and we believe that highlighting this problem in the CC community will enable necessary, and more aggressive advancements in developing computational methods for CPS.
Acknowledgements

This work is supported in part by NSF IIS 1564080, ONR N000141612835 and DARPA award W911NF-19-2-0006. The authors would also like to thank Dr. Daniel C. Dennett and Dr. Christopher G. Atkeson for their valuable feedback.

References

Creativity explained by Computational Cognitive Neuroscience

Frederic Alexandre
Inria Bordeaux Sud-Ouest; Labri UMR 5800; IMN UMR 5293
146 rue Leo Saignat; 33076 Bordeaux, France
Frederic.Alexandre@inria.fr

Abstract

Recently, models in Computational Cognitive Neuroscience (CCN) have gained a renewed interest because they could help analyze current limitations in Artificial Intelligence (AI) and propose operational ways to address them. These limitations are related to difficulties in giving a semantic grounding to manipulated concepts, in coping with high dimensionality and in managing uncertainty. In this paper, we describe the main principles and mechanisms of these models and explain that they can be directly transferred to Computational Creativity (CC), to propose operational mechanisms but also a better understanding of what creativity is.

Introduction

Artificial Intelligence (AI) has developed approaches based on knowledge manipulation and others based on data processing. The latter ones have made tremendous progresses recently, mostly due to technological improvements. Nevertheless, the development of AI remains constrained by a series of limitations, which are present from the birth of this domain and still unsolved. This is also the case for CC, seen as a subfield of AI (Colton and Wiggins 2012).

The three limitations of Artificial Intelligence

These limitations are mostly methodological, which means that they do not prevent the design of projects of AI or CC but they render difficult their development up to a realistic size and they miss a realistic description of certain characteristics, particularly when targeting their autonomy.

Giving a semantic grounding

Probably, one of the most fundamental limitations of AI is the difficulty to make sense and exploit the meaning of the objects they are manipulating. This limitation is also important in CC where a novel point of view can come from such a semantic analysis. This is often related to the incapacity to ground these objects in the real world (Harnard 1990) and to propose solutions related to the intrinsic meaning of the objects in the task. This problem is generally addressed by developing ontologies of the task, but these ontologies are often built explicitly from knowledge, whereas in humans, intrinsic meanings are generally acquired implicitly and difficult to express explicitly.

Coping with high dimensionality

Recently, the increase of computational resources allowed AI to explore tasks in high dimensional spaces but this progress remains negligible as compared to the virtually infinite size of state spaces in realistic tasks. Similarly, (Colton and Wiggins 2012) underline the growing tendency in CC to exploit web and other large resources. A strategy is to use heuristics and to explore a limited part of the state space. A side effect of long lasting exploration in high dimensional state spaces is that learning requires many examples and long training times, which is not realistic, as compared to the ability of animals and especially of humans to learn quickly with few examples.

Managing uncertainty

Tasks in the real world can be stochastic, marring observations with noise; they can be volatile, implying that a relation learned one day can suddenly be no longer valid. A related important problem, when an intelligent system receives an error signal, is to decide if this is due to stochasticity (in the case, the best strategy is to insist) or to volatility (requiring to look for a new response). This is related to the balance between exploration and exploitation, questioned in many models in AI. Exploration can be purely random, but this is not consistent with observation of a certain degree of flexibility in changing environments, where the appropriate behavior is immediately reinstated each time the same context is revisited. Choosing between exploration and exploitation and going beyond random processes are also important issues in CC. All these problems are generally tackled in various probabilistic frameworks (like bayesian techniques). Some of them propose solid and mathematically grounded methods but they are computationally prohibitive, thus adding to the problem of high dimensionality defined in the previous item.

The CCN approach

CCN models are primarily developed as a way to operationalize proposed cognitive concepts, together with their brain implementation. They can sometimes give a biological basis to classical models in AI as it is the case in Reinforcement Learning. They can also help study how the brain solves some problems that remain difficult with classical mathematical approaches. This is specifically the case with the three limitations mentioned above, as we describe now in more details.
Giving a semantic grounding  It is generally considered that access to the semantic meaning of objects is due to the fact that we have a body that we can perceive externally (sense of exteroception, including proprioception) and internally (sense of interoception (Craig 2009)). Considering the consequences of taking these sensations into account onto cognition is studied in what is called embodied AI (Pfeifer, Bongard, and Grand 2007).

Internal representation of the body is built in the insular cortex. On the one hand, rich signals of pain and pleasure allow to elaborate reinforcement signals much more complex than the unique scalar generally used in reinforcement learning. Learning to anticipate these signals (pavlovian conditioning, see a model in (Carrere and Alexandre 2015)) elicits what is called emotion and plays a pivotal role in decision making by allowing to compare different signals under a common currency, as summarized by the principle of the somatic markers (Damasio, Everitt, and Bishop 1996).

On the other hand, other important internal signals form motivations and can be related to physiological needs (extrinsic motivations) as well as needs for certain kinds of information (intrinsic motivations). Such signals are particularly important to generate internal goals and to prevent from obeying only stimuli in the environment. This is an important dimension for goal-directed behaviors, see computational frameworks in (Pezzulo and Castelfranchi 2009).

Coping with high dimensionality  One important contribution has arisen from CCN about the way humans can learn in high dimensionality with the Complementary Learning System (CLS) framework (McClelland, McNaughton, and O’Reilly 1995). In this framework, it is proposed that humans can, at the same time, learn slowly concepts in semantic memory, in the cortex, and store quickly episodes in the hippocampus. Later on, by a phenomenon called consolidation, the hippocampus will send episodes back to the cortex to train it off-line and ensure good properties to the learning process, for example to avoid mix-up, also called catastrophic forgetting. This kind of principles has been used in Machine Learning to propose ways of training large architectures (for example deep networks) with “not so big data”, cf a model in (Drumond, Viéville, and Alexandre 2019).

Recently, progresses in neuroscience about the hippocampus and in the corresponding models led to more precise explanation about information encoding in the hippocampus, and particularly about place cells and grid cells (model in (Stachenfeld, Botvinick, and Gershman 2017)) and their proposed contribution in the goal-oriented exploration of high-dimensional spaces. It has particularly been proposed that replays of episodes for consolidation are not random but obey subtle strategies (implemented in (Mattar and Daw 2018)) and that they can be organized in time to participate to the elaboration of knowledge-based strategies in the frontal cortex (cf computational framework in (Daw 2018)). It has also been experimentally shown (Derdikman and Moser 2010) that the hippocampus not only replays stored episodes but is also able to create new episodes combining pieces of old episodes, thus promoting an imaginative training with virtual episodes.

These concepts and models are today transferred to the domain of AI to elaborate what is called Episodic Reinforcement Learning (Gershman and Daw 2017) and participate to the design of more realistic and faster learning in high dimensional data spaces.

Managing uncertainty  Defining the stochastic or volatile nature of the environment has been related to the role of neuromodulation and exploited in models (Alexandre and Carrere 2016), adapting a general framework of behavior selection to the estimated kind and level of uncertainty.

As for the flexible adaptation of behavior to a volatile environment, the role of the prefrontal cortex has been mentioned for a long time in neuroscience, observing that patients with a frontal lesion demonstrate perseveration and are unable to adapt to changes (Nauta 1971). Accordingly, it has been proposed that the prefrontal cortex is the place where nondominant behaviors are defined (Wise 2008) or, to tell it differently, the place where top-down modulations are sent to other regions of the brain to insert internally generated priorities and to help resist to immediate responses driven by stimuli (Mesulam 2008), thus defining two important mechanisms in the prefrontal cortex: the inhibition of dominant and occasionally non-adapted behaviors and the triggering and maintenance by working memory of attentional focus towards characteristics supposedly important to generate the presently appropriate behavior.

Corresponding models describe the prefrontal cortex as a region where tasks in specific contexts are represented, thus defining the notion of Task Sets (Domenech and Koechlin 2015), learning in certain contexts to inhibit the default behavior and to suggest new behaviors by an attentional process (cf model in (O’Reilly et al. 2002)) biasing the current perception. In order to select the pertinent nondominant behavior, some interesting mathematical frameworks have been proposed (Collins and Koechlin 2012) to tame the combinatorial exploration and generate a limited number of hypotheses. Other interesting and bio-inspired computational mechanisms have been proposed to control and monitor the execution of such behaviors, particularly in the case of hierarchical planning (Pezzulo and Castelfranchi 2009).

These models and concepts are also presently transferred to AI in so-called Meta Reinforcement Learning (Wang et al. 2018), considering that, to adapt to the changing world, a meta learner must quickly learn to select a specialized (and slowly learning) learner, depending on the context.

Another important limitation in AI: Creativity  It is often mentioned though disputable (Boden 2009) that creativity remains one of the rare cognitive phenomena that AI cannot replicate (with other phenomena like sense of humor, not to say consciousness). Remarkably, studying solutions proposed by CCN to remedy limitations in AI (and consequently in CC), we argue (and will establish below) that they massively rely on cognitive characteristics related to creativity. We consequently propose that CCN could be pivotal to fuel CC with fresh ideas and particularly add a global view often missing in AI.
Defining creativity

Whether dealing with creativity in specific domains (like musical improvisation or scientific creativity) or from a general point of view (Beaty et al. 2016), authors often mention that creativity has two main steps: creating a novel idea and verifying that this idea is useful or appropriate to the task (Dietrich 2004); else the idea is adapted or rejected and another novel idea is elaborated. These two steps are respectively termed divergent and convergent thinking in human creativity research (Jung et al. 2013). Corresponding mechanisms in CC are called “generate and evaluate”. The first step can be associated to insight (Kounios and Beeman 2014) and originate from an emotional or a more cognitive process (Dietrich 2004). As for the second step, it is generally proposed that the assessment of the value of what has been proposed is carried out by brain circuits associated to executive functions. The brain circuits responsible for these two steps have been studied in different ways as it will be discussed in more details below.

The two steps of creativity

Concerning the divergent thinking step, it is reported that self generated thoughts can be spontaneous or goal directed (to meet specific task demands) (Dietrich 2004). This can correspond to the reinterpretation of a situation to produce a nondominant interpretation (Kounios and Beeman 2014) or to using generative models to try to explicitly build a novel solution, which has been qualified as a more restricted but more efficient solution (Dietrich 2004). (Boden 2009) proposes that novel ideas can be produced by combination, exploration or by transformation. (Dietrich 2004) proposes four types of creativity: In the first step, novelty can come from emotional or cognitive structures and the processing can be deliberate or spontaneous.

Among neuronal mechanisms evoked above, it can be remarked that the hippocampus is a good candidate to generate spontaneous novel ideas and that the mechanisms in the prefrontal cortex, responsible for selecting a new Task Set, might participate to the explicit elaboration of new thoughts. Let us also underline that the model proposed in (Collins and Koechlin 2012) explicitly mentions that if no existing Task Set is appropriate to the task, a new Task Set might be created by the combination between two existing Task Sets or by the random generation of a new one.

The convergent thinking step is often related to cognitive control, the role of which is also to ensure appropriateness in the execution of a behavior. Appropriateness can be syntactic (checking that procedural constraints are respected, which is easy to do with a generative model) or semantic (checking that manipulated objects have consistent values, which is probably less easy (Dietrich 2004; Boden 2009)). In all cases, it is proposed that these verifications are made by the PFC, with the corresponding information set in working memory for their conscious evaluation.

The two steps of creativity in brain circuits

The functional description of brain circuits is often made through networks of brain regions (Mesulam 2008), highlighting some regions as critical hubs for the spreading and processing of information. The attentional, semantic, default, cognitive control and salience networks are the major ones and play a central role in creativity (Jung et al. 2013).

The default network (activated by default, for spontaneous thinking, including parietal cortex, medial prefrontal cortex and hippocampal regions) is much involved in creativity and is known for its links to episodic memory in the hippocampus, spontaneous retrieval, replay activity and simulations based on personal past experiences (Dietrich 2004).

To evaluate the semantic and syntactic appropriateness of candidate ideas, a reference is made toward an emotional system for biological significance of events (invoking the semantic network, associated to the insula and other limbic regions) and an information processing system to perform detailed feature analysis involving attentional control networks (Beaty et al. 2016). As a synthesis, pertinence, adaptation or rejection of candidate ideas is made by the cognitive control network, involving the lateral prefrontal cortex and the anterior cingulate cortex, performing top-down modulation of self generated information for efficacy evaluation, selection and adaptation to the task (Beaty et al. 2016).

Dynamics of inhibition and excitation in hubs and networks are observed with the analysis of EEG activity in behavioral and psychological tasks (Kounios and Beeman 2014). It must be remarked that corresponding tests of creativity are in fact relying on measures of fluency, flexibility, originality and elaboration, all measures also related to problem solving. This is also a strong argument to propose that the same brain circuits and cognitive mechanisms are used for higher brain functions and for creativity (Dietrich 2004).

Discussion

CCN models presented in this paper insist on the need to build, inspired from neuroscience, different kinds of representations in different regions of the brain (Alexandre, Carrere, and Kassab 2014) and to organize the selection of behavior through the interactions between different kinds of memories (Alexandre 2000).

Understanding these mechanisms and corresponding brain circuits is particularly important to address current limitations in AI. As a central claim for this paper, we propose that these mechanisms and cognitive processes are also central to understand creativity as it is carried out in the brain. Perhaps this is not so surprising if we consider that among the mechanisms the brain has developed to circumvent the tedious and systematic analysis of some situations, creativity might be a choice mechanism to efficiently explore novel approaches.

The present paper mentions operational models that could be directly exploited to implement CC. It also proposes paths of research to implement specific mechanisms. Spontaneous processing could be related to the hippocampus and its role in the default network, whereas deliberate processing would be more frontal and related to the combination of Task Sets. The more difficult aspect of combinational creativity could be linked to the limitation of semantic grounding, thus pleading for a closer look to the semantic network. Using CCN models to implement CC could be also an easier way to choose the type of creativity to generate or even
to study fundamental differences between human and non-human creativity (Colton and Wiggins 2012).

The very nature of creativity itself can be questioned from the elements reported here. Creativity is often associated to the generation of completely novel ideas. What is reported here about convergent but also divergent thinking is that most of the time, old memories are used to create new ideas. Here also, it is the old that makes the new. What is also reported here (in line with (Boden 2009) claiming that creativity is not magic) is that CC, indeed considered as a scientific questioning, suffers mainly from fragmentation in AI research (Colton, de Mantaras, and Stock 2009) and could highly benefit from the global framework proposed by cognitive neuroscience.

References
McClelland, J. L.; McNaughton, B. L.; and O’Reilly, R. C. 1995. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. Psychological review 102(3):419–457.
Mexican International Colloquium on Computational Creativity

Rafael Pérez y Pérez
Departamento de Tecnologías de la Información, División de Ciencias de la Comunicación y Diseño
Universidad Autónoma Metropolitana, Cuajimalpa
CDMX, México
rperez@correo.cua.uam.mx www.rafaelperezyperez.com

Abstract
This paper describes the story of the Mexican International Colloquium on Computational Creativity, which has been carried out for 14 years. I describe the activities around this academic event and present a list of all invited speakers. I finish with some thoughts about the experience of organizing this Colloquium.

Introduction
The Autonomous Metropolitan University at Cuajimalpa (UAM Cuajimalpa) is a public University that mostly provides education to students that belongs to families with low incomes. One of its main goals is to promote interdisciplinary research. Currently, the University offers the only postgraduate program in the country where the fields of design, sciences of human communications and computer science converge (this is a two years, full time master course). Because computational creativity (CC) is a great example of interdisciplinary collaboration, I have been organizing for 14 years what I call the Mexican International Colloquium on Computational Creativity (http://www.rafaelperezyperez.com/coloquio-internacional-creatividad-computacional/). This academic meeting pursues five main goals:

1. To promote the study and research on CC.
2. To make our local community aware about the importance of CC in society.
3. To reflect about how technology is changing the world.
4. To learn about successful interdisciplinary projects.
5. To allow students to establish contact with top researchers in the field.

The Colloquium was held for the first time at Center for Applied Sciences at the National University (UNAM) in 2002, where I used to work. Despite its success, the authorities did not support the event and became impossible to organize a second edition. In those years, some colleagues were shocked that I was promoting the study of creativity using computers; some senior researchers at that Center strongly recommended that I focused on “normal topics that included mathematics”. Two years later my contract was not renewed.

In 2007, with the support of the authorities at UAM Cuajimalpa, my new academic home, it was feasible to return to organize the Colloquium. Since then, it has been carried out annually without interruption. For the first eight years, the Coordination of Postgraduate Studies in Computer Science and Engineering at UNAM, where I was an invited lecturer, supported the event. However, in 2015, they decided to stop sponsoring the Colloquium, expressing doubts about how useful CC research was for the students. That year, the National Council of Science and Technology in México (CONACYT) funded my research project in CC that included financial support to continue organizing the Colloquium. The Division of Design and Science of Human Communication at UAM Cuajimalpa also backed the event.

Researchers from Europe, the United States and Latin America have participated in this academic reunion. We feel lucky that we have been able to listen to recognized scientists in computational creativity and related areas. In the next sections I explain the general organization of the meeting and provide some thoughts about the experience of organizing fourteen editions of this Colloquium.

The Colloquium
The Colloquium takes place once a year at the end of October or beginning of November, coinciding with the famous Mexican celebration of the Day of Death.

Through all these years, 28 top researchers in CC and related areas have partaken in the event; Pablo Gervás has the record of more participations with four contributions, followed by Nick Montfort with three contributions, Mike Sharples and Michael Mateas with two, and the rest with one participation (see Table 1).

The Colloquium consists of a series of master lectures given by the invited speakers. Each year, one theme is chosen as the focus of discussion: automatic writing, automatic
design, interactive systems, learning, emotional music generation, painting, interdisciplinary collaboration, social concerns, cognitive approaches, videogames, language, creative collaboration, e-literature, inclusive AI.

Thus, the perspectives, approaches, methodologies, an areas of research exposed through all these years have been vast. The reader can watch some of these lectures in the web page of the Colloquium.

<table>
<thead>
<tr>
<th>Year</th>
<th>Topic</th>
<th>Guests</th>
</tr>
</thead>
<tbody>
<tr>
<td>2002</td>
<td>Creativity in writing</td>
<td>Mike Sharples</td>
</tr>
</tbody>
</table>
| 2007 | Design | John Gero  
Pablo Gervás  
Paula Montoya |
| 2008 | Interactive systems | Pablo Gervás  
Brian Magerko  
Nick Montfort  
Mark Riedl |
| 2009 | Automatic narrative generation in learning | Mike Sharples  
Liz Beaty  
Pablo Gervás  
Michael Mateas |
| 2010 | CC as an example of interdisciplinary collaboration | Graeme Ritchie  
Dan Ventura  
Nick Montfort |
| 2011 | Painting and Games research | Simon Colton  
Michael Young |
| 2012 | Music | Amílcar Cardoso |
| 2013 | Social concerns | Sneha Veeragoudar Harrell  
Fox Harrell  
Nick Montfort |
| 2014 | Cognitive perspectives | Mark Turner  
Tony Veale  
Geraint Wiggins |
| 2015 | Videogames | Clara Fernández Vara  
Pablo Gervás  
Michael Mateas |
| 2016 | Language | Anna Jordanous  
Paolo Rosso  
Juan Manuel Torres |
| 2017 | Creative collaboration with intelligent systems that compose music | Maya Ackerman |
| 2018 | e-Literatures | Alex Saum  
Carlos León |
| 2019 | Inclusive literature in the era of AI | Milton Läufer  
Mario Silva |

<table>
<thead>
<tr>
<th>Year</th>
<th>Name</th>
<th>Title</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013</td>
<td>Nick Montfort</td>
<td>Exploratory Programming Workshop</td>
<td>6</td>
</tr>
<tr>
<td>2015</td>
<td>Clara Fernández Vara</td>
<td>Creating Fictional Worlds for Games</td>
<td>8</td>
</tr>
<tr>
<td>2016</td>
<td>Anna Jordanous</td>
<td>Making computers that can communicate creatively</td>
<td>3</td>
</tr>
<tr>
<td>2016</td>
<td>Paolo Rosso</td>
<td>Automatic analysis of user profiles on social networks</td>
<td>3</td>
</tr>
<tr>
<td>2017</td>
<td>Maya Ackerman</td>
<td>The Computer as a Creative Partner</td>
<td>3</td>
</tr>
<tr>
<td>2018</td>
<td>Alex Saum</td>
<td>Home electronic literature: tools to develop computational creativity in our daily practices</td>
<td>3</td>
</tr>
<tr>
<td>2018</td>
<td>Carlos León</td>
<td>Experiencing computational creativity: generating stories just like a computer</td>
<td>3</td>
</tr>
<tr>
<td>2019</td>
<td>Mario Silva</td>
<td>Literary workshop: “Letters with Smoke and Sweet”</td>
<td>1.5</td>
</tr>
<tr>
<td>2019</td>
<td>Milton Läufer</td>
<td>A practical approach to computational literature</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Table 1. List of speakers at the Colloquium.

Table 2. List of workshops given at the Colloquium.

Figure 1. Students participating in a workshop taught by Carlos León.
Meetings with postgraduate students to discuss their projects and get feedback from the guests also take place.

Additionally, the Colloquium promotes the development of practical skills. In 2002 Mike Sharples offered a workshop on creative writing. In 2013, Nick Montfort organized a workshop about exploratory programming for those in the arts and the humanities. From 2015 onwards, we always have included at least one workshop whose main purpose is to consolidate the knowledge and concepts discussed during the keynotes (see table 2). All they have been very successful. Once we got 60 attendees! No doubt that to put in practice what has been discussed in the lectures has proved to be important experience for our students. However, from my perspective, the opportunity to have a close and informal interaction with top researchers has been more significant (see figure 1).

Most of the invited speakers do not speak Spanish and most of our students do not speak English. We have hired simultaneous translation to deal with this issue. The solution has worked although it is evident that language is an important factor in order to establish a good connection with the audience.

Some female students have expressed how the presence of women as keynote speakers have inspired them. I particularly remember this shy student who hardly expressed any emotion. When Clara Fernández Vara’s workshop ended, the student’s whole face was covered by this huge beautiful smile; she expressed an enormous happiness. Something changed that day. So, it is my belief that the Colloquium has contributed to the personal development of many of those who have attended it. This has been my main motivation during all these years.

As part of the dynamic generated by the Colloquium, in 2011 we organized the International Conference on Computational Creativity. And in 2015 we published in Spanish what I believe is the only book about computational creativity written in a language other than English. The idea came up during the 2014 meeting. The book includes nine chapters, all written by authors that have participated in the Colloquium. All these results have been very satisfactory. However, there are still many challenges ahead of us. Although many of our students express curiosity and at times look impressed by the products generated by creative agents, it seems that they do not see this area as a field to cultivate. For instance, few undergraduates and postgraduates deliver dissertations in topics related to CC. They prefer to work in more “profitable” areas that they believe will give them better opportunities to get a good job when they finish their studies.

Similarly, with some notable exceptions, it has been really difficult for me to convince my colleagues about the benefits that this field offers for their research interests.

Based on these experiences, I would like to suggest to the CC community the discussion of the following points:

- Besides activities like academic events, a decisive endorsement from the university’s authorities is vital to promote the area. Financial support, like the one this Colloquium has received, is essential but not enough. An important step is to incorporate into the curricula classes where CC is discussed. Tutors from diverse backgrounds need to be instructed about the importance of CC in their respective fields; in this way, they might feel more comfortable including such topics in their lectures and seminars.

- The CC community must make an effort to improve its communication with people outside academia. We must learn how to broadly show, for example, the usefulness that developments based on creative systems might have in areas like education, entertainment and the private industry. Likewise, it is necessary to communicate more effectively how research on computational creativity contributes to the understanding of the digital revolution and its effects on society. That is, we must clearly illustrate to general audiences the significance of learning about CC.

- Similarly, the Association for Computational Creativity should take a much more active role to promote the field. Particularly, in those communities outside Europe and the USA. For instance, I suggest that the Association develop and distribute, in different languages, material that instructors from different backgrounds could easily integrate in their classes. I believe that these kind of actions will have a strong positive impact. I would like to go a little further. Given that one of the objectives of the CC community, and of science in general, is the dissemination of knowledge, and because we are creative people, we must promote the development of educational material that is useful to groups with limited technological resources and who do not necessarily speak English. In that way, we will become more inclusive.

I believe that discussing and addressing these points will benefit the whole CC community.
The near future of the Colloquium is unclear. I have not been able to find someone to take over from the organization. Because of the pandemic, the 15th edition of the Colloquium has been postponed. To make things worse, the current federal government is cutting the financial support for universities, which probably will force my Institution to stop funding academic events like this one. So, probably the Colloquium is facing its last moments.

All in all the organization of the Colloquium has been a wonderful and satisfactory adventure (see figure 2). I hope this paper encourages others to organize similar events. It would be great to share experiences.

One last thought. The generosity of all those researchers that have come to share their knowledge and experience with our students is enormous. I would like to express my eternal gratitude to all them.

Acknowledgements

The Mexican International Colloquium on Computational Creativity has been sponsored by the Autonomous Metropolitan University at Cuajimalpa, the National Council of Science and Technology in México (CONACyT), and the Coordination of Postgraduate Studies in Computer Science and Engineering at UNAM.
Abstract
Machines are used to be well-known for doing repetitive jobs, following code produced by humans, so not often imagined as capable of producing unexpected results without direct human intervention in each unique case. The generative art broke this boundary by incorporating random variables extracted from a natural phenomenon into the coded environment. Using this method, our project introduces physical laws into 3D printing instructions; a custom slicer and G-Code compiler were developed, allowing manipulation of Z-heights of extrusions from the 3D printer so gravity can be used to generate randomness in 3D outputs. As a result, unique organic and porous forms could be created from the 3D printing process, differing from traditional, watertight solid prints.

Introduction
Each of nature's products is different. No two objects, even those formed in similar environments, are the same. As humans, we learn from nature and imitate nature, and then we create machines to mimic our actions autonomously and do jobs for us. However, so often, the uniqueness that occurs natural formation processes disappears when objects are made within a coded, controlled environment designed for identical reproduction. Every manufactured object comes in smooth and air-tight surfaces, which is different from what nature creates.

Buckminster Fuller observed that "... there are no solids in the universe. There is not even a suggestion of a solid. There are no absolute continuums. There are no surfaces. There are no straight lines." The authors of this work were drawn together by our shared interest in porosity and nonuniformity, as shown in a natural example, stalagmites; developed a method to hack a 3D printer to print in a way that resembles natural processes. Conventional 3D printing "best practices" encourage the designer to generate "watertight solids" for error-free slicing and durable fabrication. In this project, we explore an alternative form of G-Code (ISO 6983-1:2009) generation, manipulating the starting Z-height of the extrusion point so gravity can create randomness within the rigidly coded environment. We hypothesize that the randomness will mimic nature's surfaces and forms; the visual of mingled wiggly lines stacked chaotically almost resembles the surface of a cotton ball or cocoon. Individual prints from the same code will be unique.

Randomness and Creativity
According to Colton and Wiggins (2012), the definition of Computational Creativity research is 'The philosophy, science, and engineering of computational systems which by taking on particular responsibilities, exhibit behaviors that unbiased observers would deem to be creative.' People tend to attribute any creative outcomes resulting from computations to human coders, and the human has a constant predilection in their choice. Hence, the authors state that the coder role should remain as an unbiased observer for truly computational creativity.

The question is, 'Although the coders do not put any bias into the code, what makes the machine creative if all the outcome is the same all the time?' There are no unreasonable components in a machine's calculations; it does not produce any noble outcomes that differ from others so that they can have values. Thus, we suggest using randomness is one of the solutions to represent computational creativity that stays out of humans' biases effectively while it still creates unique outcomes.

Using Physical Properties to Generate Randomness
Random numbers are typically generated through HRNG (hardware random number generator) or TRNG (true random number generator), programs commonly used in gambling, juror selection, and military draft lotteries. These devices use physical phenomena as variables operating independently of bias from the human coder. Frequently, noise signals from thermal or quantum phenomena are used for this purpose.

Stalagmites
Stalagmites form on cave floors through the accumulation of material dripping from above. Thus, gravity plays an
essential role in this development, generating unique forms due to differences in the heights between cave ceilings and floors and variable material properties (lava, minerals, mud, peat, pitch, or sand, among others). Our choice of gravity as the physical variable mimics these biological formations; it suits the creation of 3D objects as we desired.

Related Works

The notion of superseding deterministic algorithms and harnessing randomness to create novel outcomes has been previously explored in multiple domains. Rashel and Manurung (2014) developed an Indonesian poetry generator with a combination of random factors to generate diversity. Similarly, Tomasić, Žnidaršič, and Papa (2014) created a random slogan generator to enhance brainstorming process. However, both focus on algorithms for order and choice of words; it seems distant from using nature variables and creating 3D as an outcome. Michael and Simon (2015) extended their research to video games' contents and colors, but they state that random features limit the machines' ability to make intelligent choices rather than promoting it; thus, they focused on generating preference code, which is made through subjective computation.

In that sense, the generation of novel 3D objects that can be 3D printed is similar. Joel, Sebastian, and Jeff (2016) use an advanced algorithm that is not bounded by strict mathematical coding and creates more arbitrary outputs. However, the work concentrates on the translation of 2D image as a base of the creation to smooth 3D objects using a deep neural network, and do not introduce any physical properties as randomness creating agents to give variance in 3D printed results. Claire, AlOthman, and García del Castillo y López (2018) propose a method to 3D print large spatial lattices of porous clay structures. However, as opposed to using this method to generate diverse objects, the authors incorporate real-time self-recalibration mechanisms to counter the unpredictability of the material deposition.

Printing Error

The PRINTING ERROR slicing tool helps the user generate patterns and translate them into 3D forms. This tool is created using Grasshopper, a Rhinoceros 3D CAD (Computer-Aided Design) software's visual programming plug-in. The user can then experiment with the distance between layers, introducing an element of chaos into the print— though the PLA (Polylactic Acid) filament begins to curl if the distance is too high. The slicer helps familiarize the user with a new set of tolerances and practices, such as beginning and ending a fragile experimental print with a tight structure to prevent it from unraveling.

Slicing

When a 3D model is sent to a 3D printer, the printer's software automatically slices the object in the XY plane with about 0.3mm gaps between each line, the thickness of the melted PLA filament. This results in a watertight solid. We hypothesize that if the gaps between lines increase, the melted PLA filament will fall below the layer unpredictably. To manipulate the Z-height of the gap, we developed our slicing algorithms, incorporating custom patterns that add spatial movements of the print head in the three dimensions.

With the slicer, users can input their desired 3D form to extract lines on XY planes. Users can then determine the number of layers by typing numbers they want.

Pattern - Distortion

The adjacent layers of extracted lines can be grouped. Users can select different patterns for each layer or group. Each layer will contain one continuous line, representing the 3D printer nozzle route, and will be divided into number segments, then distorted in angles to make a pattern.

3D Print

After lines are converted into desired patterns, they are translated into G-Code. It contains instructions for 3D printers, including not only the coordinates for shapes, but also nozzle printing speed, nozzle temperature, and bed temperature, which are important factors that affect outcomes. The extent might differ depending on the type of printers, but users can adjust those factors.

Figure 1: This is a simulation of the G-Codes generated by the PRINTING ERROR slicing tool. The original 3D form was a sphere.

Z-height of Extrusion

There was a limitation on the maximum z-height of each gap between layers; typically, the gap exceeding 1.8mm counted as an error, and the printer stopped printing.
Figure 2: The 3D printer builds up layers with different heights, creating unexpected wobbly shapes.

**Printing Speed**
Printing speed was one of the critical aspects to consider. Slower extrusion rate resulted in increased PLA filament curling and created more chaos in the outcome.

**Nozzle Temperature**
The nozzle temperature needed to be set up differently according to types of PLA filament but typically ranged between 260-310 °F. The higher temperature of extruded PLA filament tends to bond with the bottom layer well.

**Bed Temperature**
Lack of a supporting structure on the bottom and low-density structure causes the object to move around easily while printing. Thus, higher bed temperature was essential for it to stick to the printing bed firmly.

Figure 3: Low bed temperature made models not stick to the ground, so they moved and shifted while printing.

**Different Types of PLA filament**
When experimenting with our slicer, decay was on our mind: we printed a series of objects using biodegradable wood- and algae-based PLA filament, pursuing forms that would be strong enough to retain soil or serve as scaffolding for growing mycelium, but fragile enough to succumb to biodegradation over time. We were drawn to the transparency of these prints and began a second series using semi-clear and clear PLA filament, investigating translucent effects.

Figure 4 (From the top left to bottom right, made out of wood-based PLA, biodegradable algae PLA, Stone-like texture PLA, Semi-translucent PLA): Different types of PLA filaments demonstrate unique characteristics and different potential applications.

**Challenges**
There are several difficulties in 3D printing. Often gaps between the layers were counted as an error, so it was not computed in a 3D printer. The curled-up extruded PLA filament casually picked up by the moving nozzle. Even if it passes all those problems, without supporting structure that holds lines, the object gets unraveled easily. All patterns need a tight structure to hold them together.

Figure 5: This model is composed of two different structures, loose and tight; the tight structure to solve the unraveling problem.

**Results**
Aside from conventional 3D printing best practices that encourage watertight solids using stable fabrication methods, we hoped to explore how far we could coax the printers to stray from solidity, opacity, and durability retaining structural integrity. The most exciting and effective results came from manipulating the Z-height. The tool, therefore, encourages users to experiment with the distance in between.
**Figure 6:** Full customization is possible, but 18 different tested patterns are provided as a starter. Users can simply plug-in or mix-match to their desired 3D form.

**Application**

Nature-friendly materials, like wood or algae-based PLA filaments, show a rougher surface with extra-porosities that allow plants and other organic species to stick; they have a high potential as a green building structure.

Each product's uniqueness could be shined if compared with other manufactured products due to its exclusiveness. Using different types of PLA filament, the products will have different use-cases. However, transparent or semi-translucent PLA filaments, show excellency in smearing lights through in an ambient way, which suits to lighting designs.

**Figure 7:** The top view of 3D printed objects with various layers of patterns and manipulated Z-height extrusion.

**Conclusion**

This experiment introduces randomness to machine processes by introducing physical properties in 3D printing, creating unique objects through hard coding. Inspired by the formation of unique stalagmites in caves, the process sought to explore nature's randomness in machine form. By manipulating Z heights of 3D printer extrusion points, we created unexpected organic forms that differ from regular watertight solids. Our project aims to show alternatives to current factory manufacturing methods that produce identical objects and destroy uniqueness. For us, creativity mimics what nature creates and how nature creates. We integrated gravity, a natural law, into coded instructions for 3-D printing to introduce this unique creativity into the computational world.

**Acknowledgments**

This project encountered many challenges on the way to our desired outcomes on the table since it was experimental in nature. We would like to express our appreciation to Daniel Tish for his guidance.

**References**


Using Adaption-Innovation Theory to Simulate Robustness in Design Teams

Noriana Radwan¹, Christopher McComb²

¹Department of Industrial and Manufacturing Engineering, ²School of Engineering Design, Technical and Professional Programs The Pennsylvania State University
University Park, PA
nxr406@psu.edu, mccomb@psu.edu

Abstract
Creative design is often accomplished through teams, and the specific composition of those teams can limit or enhance the sum creativity. However, it is not known how team composition is related to a team’s ability to achieve good solutions over various problems (i.e., robustness). Here, we build that relationship between composition and robustness through a series of agent-based simulations. The factor that we specifically investigate is cognitive style, which describes the manner in which individuals solve problems and present solutions in social interactions based on cognitive processes. Under Kirton’s Adaption Innovation (KAI) Theory, cognitive style is related to creativity. Specifically, an individual’s KAI score, the defining measure of cognitive style, describes the degree to which they prefer high-utility solutions, or high-novelty solutions – the two necessary conditions for creativity. In many cases the long-term success of a team is closely tied to their ability to perform consistently well across multiple design problems, termed robustness. Leveraging computational agents, we use adaption-innovation theory as the primary factor to examine robustness among homogenous and heterogeneous agents. Different approaches to composing teams with homogenous and heterogeneous cognitive styles did not substantially impact robustness. However, the average robustness of the teams improved as team size increased.

Introduction
For teams to be effective, we expect members to be able to collaborate with each other in sharing ideas and offering support to achieve their goals. However, as teams are often composed of individuals from diverse backgrounds and skill sets, issues may arise that create differences and conflict within the team. While the divergence of opinions can simulate creative ideas and solutions (Chen, 2006), it can also be detrimental to the design process, and present solutions in social interactions based on cognitive processes. Under Kirton’s Adaption Innovation (KAI) theory refers to cognitive style in which the manner of accomplishing cognitive tasks can be placed on a continuum (Jablokow, 2000), with the extremely “adaptive” and extremely “innovative” thinker are on either end of the spectrum (Bobic, 2000), with the extremely “adaptive” and extremely “innovative” thinker feel constrained by rules, cutting across paradigms close to the “status quo” and making incremental changes (Samuel & Jablokow, 2011). In contrast, an “innovative” thinker feels constrained by rules, cutting across paradigms and the existing structure to solve problems “differently”, with less concern for group consensus to achieve improved solutions, sticking close to the “status quo” and making incremental changes (Samuel & Jablokow, 2011). While team performance can be attributed to a number of factors, the examination of cognitive style offers one avenue that is particularly relevant for creativity in design.

In order to study the composition of human teams with varying cognitive styles, we use the Python implementation of the KAI Agent-Based Organizational Optimization Model (KABOOM) (Lapp et al., 2019b) to generate and investigate teams of computational agents with varying simulated styles. The KABOOM framework enables investigations of the impact of KAI Theory on teamwork. In this framework, autonomous agents with various cognitive styles interact to solve a diverse set of problems and maximize the objective function, indicating performance (Lapp et al., 2019b). KAI
is measured, in both humans and agents, with three subscores for Sufficiency of Originality, Efficiency, and Rule Group Conformity, which are the determinants of cognitive style (Lapp et al., 2019b). Sufficiency of originality is of particular importance, as it defines the degree to which individuals prefer high-utility solutions, or high-novelty solutions – the two necessary conditions for creativity.

We evaluate the performance of homogenous agent-based teams (in which each member of the team has the same total KAI score) and modify team composition to form various heterogeneous agent-based teams to identify the best combination of KAI scores to maximize robustness. Specifically, this research seeks to answer the following questions: (1) In homogenous teams where each agent has the same KAI score, which cognitive style is the most robust? (2) Which heterogeneous team composition provides the best robustness? (3) How does the size of the team affect its robustness?

Methodology

In KABOOM, the goal of the agents is to maximize their objective function using a simulated annealing optimization algorithm in which the agents explore widespread solutions from a highly stochastic approach gradually changing to a more downhill search. This reflects the nature of human problem solving (Cagan & Kotovsky, 1997). In KABOOM, heterogeneous agents possess unique cognitive styles that modify their exploration of the solution space.

In this paper, a solution is a set of parameters that define a position in the solution space, and the quality of a solution is the value of the objective function for those parameters. Team performance is taken to be the best solution any individual on a team has found.

The Problem Set

This paper implements an abstract mathematical objective function, or design problem, that can be tuned and scaled in predictable ways. It is represented by a scalar objective function \( f(x) \) of \( n \) dimensions (variables). The objective function used is a summation of a quadratic function and a sinusoidal function in the form:

\[
 f(\vec{x}) = \sum_{i=1}^{n} \alpha \cos\left(\frac{\omega \vec{x}_{i}}{\beta}\right) - C \left(\frac{\vec{x}_{i}}{\beta}\right)^{2} \text{ for } -0.5 \leq x_{i} \leq 0.5
\]

This function is varied in two ways: (1) by scaling the independent variables in all dimensions using the scaling parameter, \( \beta \), and (2) by scaling the oscillation amplitude of the sinusoid, \( \alpha \). The first parameter affects the size of the search space, while the second parameter affects the amplitude of the sinusoid. By varying these parameters, we create 25 unique design problems for running our simulations that may favor different cognitive styles.

According to past research from Lapp et al. (2019), adaptive agents (those with lower KAI scores) are hypothesized to best solve problems with a high oscillation amplitude and smaller search space. Innovate agents (those with higher KAI scores) are therefore more suited to solve problems with a smaller amplitude and a broader search space. By assessing the performance of a team with a specific composition across each of these problems, it becomes possible to assess robustness.

Team Composition Strategies

To evaluate the effects of team composition on robustness, we formed teams using three composition strategies. In the first, referred to as organic composition, teams were generated by selecting individuals with KAI scores corresponding to the distribution of scores observed in the general population (mean=92.93, std=18.20). Second, homogenous teams were each comprised of agents with the same KAI score. The seven KAI scores were linearly spaced from 60 to 140. Third, heterogeneous teams were produced by sampling from the uniform distribution of the complete range of scores. This selection was iteratively narrowed to more mid-range scores to produce a variety of team compositions.

Quantifying Robustness

At the end of each simulation trial, the team’s performance is the solution quality of the best solution any agent has had at any time during the simulation, quantified as a numerical value. We examine three values computed from the solution vector. These include the median value of each team, the lower quartile or 25th percentile performance, and the worst score to determine the lower bound on the team’s performance. The team’s “worst” score does not include outliers.

As the teams are solving a minimization function, lower values indicate a better performance. Thus, higher values indicate a worse score.

Results

Figure 1 displays the results of homogenous team simulation for six agents per team. Results from the simulation with both two and ten agents per team yield similar results.
These results indicate that among each of the teams, performance remained the same, and we see no clear “superior” KAI score. There were only minor differences between the team’s performances.

Therefore, it is not clear whether there is a KAI score that is the most robust and resilient for team composition, as there is very little difference in aggregate robustness. Among the range of KAI scores from 60 to 140, no team significantly outperformed another. Additionally, the median and 25th percentile scores followed closely to the baseline of the normal distribution, indicating that forming teams with homogenous agents yields similar outcomes to organic teams. This aligns with earlier work on cognitive style, in which it has been indicated that different cognitive styles are not necessarily better than one another, simply different (Hammerschmidt, 1996; Jablokow, 2000; Lapp, Jablokow, & McComb, 2019a; Lapp et al., 2019b). Additionally, while the overall KAI score was the same for each member of the team, subscores varied. Two agents with the same KAI values may have slightly different SO scores, affecting their creativity and preferred problem-solving approach. While team performance did not have a significant effect between teams, each of the scores improved as the team sizes increased from two to ten agents per team.

Figure 2 indicates the results of heterogeneous team simulation for six agents per team. Results from the simulation with both two and ten agents per team yield similar results.

**Figure 2.** Heterogeneous team performances with 6 agents per team. Lower values indicate better performance. Error bars represent absolute deviation. Dashed lines represent average values achieved by organic teams.

Unlike the homogenous teams, our heterogeneous teams performed slightly better than the organic teams. This may be an indication that utilizing the full distribution of scores may be advantageous, as some design problems may favor the highly adaptive or highly innovative individuals, producing better scores. There might also be some advantages to using a uniform distribution as opposed to a normal distribution for team selection, as uniform distribution allows an equal opportunity for any score to be selected. However, once again, there is not any significant differentiation in performance according to team composition, further supporting claims by Lapp et al. (2019b).

Kirton also claims that a cognitive gap, or the difference in cognitive style between individuals will have a “just noticeable difference” at 10 points, and gaps of 20 points or more will lead to significant problems between teammates (Jablokow & Booth, 2006). Thus, while both the 100 ± 5 and 100 ± 10 team are heterogeneous in that they are comprised of members with different KAI scores, there may not be enough variation in cognitive style to present them as purely heterogeneous.

To investigate how the size of the team may impact its performance, we performed the simulation to study the robustness from two-agent teams to 20 agents per team. The graphical representation of these scores can be found in figure 3.

**Figure 3.** Average performance of the teams for the median, 25th percentile and worst scores for homogenous teams.

We now observe that better scores are positively correlated to larger teams for homogenous teams. However, it should be noted that this result depends on the ability of large teams to function cohesively without hierarchy (which can be a barrier to performance). In practice, teams often form implicit hierarchical structures when larger than six members (Maier, DeFranco, & McComb, 2019). The effect of team size on heterogeneous teams yielded a similar output. We can also determine that mid-size teams may be more robust than smaller ones, but increasing team size beyond these values may not continue to yield significantly better performances. These findings are commensurate with prior research results on team size and performance, where individuals in teams collaborated with more increasing team size, resulting in larger teams outperforming smaller ones (Mao, Mason, Suri, & Watts, 2016). In theory, larger teams allow for more diversity and opportunities to communicate with and learn from others, accelerating the process of converging to a solution (Marschak & Radner, 1972).

**Conclusion**

Teams that are able to perform consistently well across various problems are considered robust. The quality of
robustness is vital, as it enables teams to be successful and work collaboratively to achieve their goals. There exist prior models of computational creativity that have assessed the way in which existing knowledge is combined to achieve novel solutions (Guzdial, Liao, Shah, & Riedl, 2018). In this research, we utilized the KABOOM model to investigate how creativity can affect a team’s overall robustness, by tuning the composition of the team based on KAI value and evaluating performance based on 25 individual problems.

The KABOOM model focuses specifically on cognitive style and does not create a comprehensive representation of team problem solving, including coping behaviors and team strategy. While the results are specific to the present simulation paradigm, this work presents compelling results that indicate how individual’s problem-solving behaviors may affect a team’s performance. Future work may involve conducting more detailed simulations as they relate to the intricacies of human problem solving, and further validating results with human subjects research.

References
Limits Theorems for Creativity with Intentionality

Lav R. Varshney†,‡
†University of Illinois at Urbana-Champaign, Urbana, IL, USA
‡Salesforce Research, Palo Alto, CA, USA
varshney@illinois.edu

Abstract

Creativity is the generation of an artifact that is judged to be novel and high-quality. Several computational creativity systems with diverse algorithmic foundations are now meeting standards of novelty and quality, as judged by experts in creative fields. The existence of these myriad design approaches suggest a natural question analogous to the one addressed in information theory: Are there fundamental limits to creativity? Here, we first review a recent mathematical formalism that captures key aspects of combinatorial creativity and yields fundamental tradeoffs between novelty and quality. The fundamental limit resembles Shannon’s capacity-cost function. Then we extend the theory to capture intentionality in creativity, treating it as a communication problem, where a creative artifact must not only be novel and high-quality, but also reliably convey a message of given information rate. The resulting fundamental limit resembles the information bottleneck optimization in machine learning.

Introduction

Computational creativity systems are now able to produce ideas and artifacts that are judged to meet standards of novelty and utility by experts in creative domains, see e.g. (Boden 2004; 2015; Colton and Wiggins 2012; Varshney et al. 2019; Keskar et al. 2019), according to a variety of assessment criteria (Jordanous 2012; Colton et al. 2014; Lamb, Brown, and Clarke 2018; Riedl 2015; Hashimoto, Zhang, and Liang 2019). Yet, it has been unclear whether there are upper bounds on how creative any system can be, whether human, machine, or hybrid. This suggests the need for a general theory of creativity that would yield fundamental limits, analogous to the Shannon limit for reliable communication in the presence of noise (Shannon 1948) or the Carnot limit for efficiency of engines (Carnot 1824). Note that such limit theorems are prevalent in mathematical systems theories (Auyang 2004) and determined within closed deductive systems that require abstraction to establish.

Fundamental limit theorems serve different several purposes. First, they establish which resources and performance criteria are fundamental and which are largely unimportant. Second, they demarcate what is possible from what is impossible, providing design insights into operating at the boundary, that is, principles for optimal designs. Third, they define fundamental benchmarks that allow an evaluation of new creativity algorithms on an absolute scale, rather than only compared to people or previous technologies. Finally, they state ideals for pushing people to build technologies that approach/achieve these absolute limits. Note that the kinds of informational fundamental limits we consider here do not take computational complexity into account.

A basis for such a general mathematical systems theory was given in prior work that formalized the structure of conceptual spaces and computational creativity (Wiggins 2006; Ritchie 2007; 2012; Hung and Choy 2013; Velardo and Vallati 2016). We recently extended this theory to include a statistical dimension, which further enabled a characterization of what is possible and what is impossible in designing computational creativity systems. The result detailed in (Varshney 2019a) provided a limit theorem on the trade-off between novelty and quality for a given creative domain. The result focused on combinatorial creativity but also captured transformational creativity as a wholly different kind of creativity. This previous work, however, only considered the technical problem of creativity and excluded consideration of intentionality—a (human) intent, inspiration, or desire to express something (Collingwood 1938; Hertzmann 2018; 2020). As described there, “These aspects of intent in creativity are irrelevant to the engineering problem”. Consideration of intentionality is also absent in previous theories of creativity (Wiggins 2006; Ritchie 2007; 2012; Hung and Choy 2013; Velardo and Vallati 2016).

Yet, it is said—especially in the Western tradition following Romanticism (but see criticisms, e.g. (Wimsatt and Beardsley 1946))—that communication of meaning in art is necessary for eliciting an aesthetic experience (Csikszentmihalyi and Robinson 1990; Cilliers 1998; Ritchie 2007). For example considering narration or poetry, (linguistic) meaning is the relation between a linguistic form and communicative intent, where communicative intents are about things that are outside of language. Communicative intent is distinct from standing meaning, which is constant across all of its possible contexts of use (Bender and Koller 2020).

Recent surveys further indicate that people want not just novelty/quality, but also intentionality and autonomy, to attribute creativity to an artificial system (Ventura 2019). These further layers of desiderata for creative systems are redolent of the three layers of communication put forth by...
Warren Weaver (Shannon and Weaver 1949):

- the technical problem (How accurately can the symbols of communication be transmitted?),
- the semantic problem (How precisely do transmitted symbols convey the desired meaning?), and
- the effectiveness problem (How effectively does received meaning affect conduct in the desired way?).

Beyond the technical problem of creativity from prior work (Varshney 2019a), here we are concerned with incorporating intentionality to consider the semantic problem of creativity.

Meaning and understanding have long been described as a key to intelligence (Bender and Koller 2020). Intention is realized when the system produces an artifact with the goal of communicating a particular message and the observer reliably understands that message from the artifact (Ventura 2019). As such, here, we aim to extend (Varshney 2019a) to include an intentionality layer, by requiring a creative artifact to not only be novel and high-quality, but also reliably convey a message.

The remainder of the paper is organized as follows. In the next section, we summarize our past theoretical framework. Next we extend that framework by considering the communicative intent of intentionality. Finally, we conclude.

**Review of Mathematical Formalisms and Limits of Creativity**

We first review the existing formalism and resultant limit theorems for the technical problem of combinatorial creativity. For brevity, here we restrict to finite artifacts, but see (Varshney 2019a) for extensions. To facilitate exposition of the mathematical formalism, we use culinary creativity as a running example.

The basic idea is to formalize the conceptual space for combinatorial creativity, to formalize the two dimensions of merit (novelty and quality), and to formalize a general description of a computational creativity algorithm that operates in the conceptual space to optimize the tradeoff between novelty and quality.

**Definition 1.** A component is an atomic unit in the creative domain, drawn from the set $\Omega$, from which artifacts are constructed.

In culinary, this would be the list of possible ingredients that people eat.

**Definition 2.** A discrete artifact is an unordered combinatorial object $\alpha$ selected from the power set $2^\Omega$ of possible components, $\Omega$, that define the creative domain.

In culinary, this would be a specific recipe, expressed in terms of its ingredients (and not amounts or instructions).\(^1\)

**Definition 3.** The known set is a set of artifacts that are already known in the creative domain, $\Theta$, also called the inspiration set. In the discrete case, $\Theta \subseteq 2^\Omega \subseteq 2^{2^\Omega}$.

---

1For brevity, we focus on ingredient lists, but extensions to structured objects like sequential recipes (which may be represented as directed acyclic graphs) or music compositions with non-commutative novelty and quality functions follow directly.

---

In culinary, this may be all recipes in published cookbooks, or recipes a given person has cooked/eaten.

**Definition 4.** Novelty is determined using a non-negative function, in the discrete case, $s : 2^\Omega \times 2^{2^\Omega} \mapsto \mathbb{R}_+$ that measures the surprise of a given artifact $\alpha_0$ in the presence of a known set $\Theta$.

A particular novelty measure that was considered is the empirical Bayesian surprise (Itti and Baldi 2006; 2009; Baldi and Itti 2010; Varshney 2019b). If we let the prior probability distribution of artifacts in the known set $\Theta$ be $P_\theta$, the creation of a new artifact $\alpha$ will update it to a posterior distribution $P_{\theta|\alpha}$. Then the Bayesian surprise $s(\alpha, \Theta)$ for a given artifact with respect to the inspiration set $\Theta$ is

$$s(\alpha, \Theta) = \int_\Theta P_{\theta|\alpha} \log \frac{P_{\theta|\alpha}}{P_\theta} d\theta.$$  

Notice that such a Bayesian notion of surprise is an expectation-based novelty measure (Grace and Maher 2019) with respect to the prior $P_\theta$. When there is lack of absolute continuity, we have an infinite value that indicates transformational creativity in a kind of hierarchy (Wiggins 2019).

**Definition 5.** Quality is determined using a non-negative function $q : 2^\Omega \mapsto \mathbb{R}_+$ that measures the utility of a given discrete artifact $\alpha_0$.

In culinary, quality may be a measure of flavor derived from the hedonic psychophysics of olfactory perception.

**Definition 6.** A creativity algorithm $G$ is a probabilistic process $P_A(\alpha)$ that produces a set of $n$ artifacts $\{\alpha_i\}_{i=1}^n$.

This view of creativity as stochastic sampling in the conceptual space is very general. In degenerate forms, such a definition encompasses generative algorithms that enumerate the entire space or optimization algorithms that directly and deterministically generate just $n = 1$ possibility.

With this formalism in place, the fundamental tradeoff between the average quality and average surprise produced by sampling algorithm $P_A(\alpha)$ is cast as follows.

$$S(Q) = \max_{P_A(\alpha)} \mathbb{E}[s(A, \Theta)].$$

This is not only a limit theorem, but also implies an optimal creativity algorithm, the extremal $P^*_A(\alpha)$,

$$P^*_A(\alpha) = \arg\max_{P_A(\alpha)} \mathbb{E}[s(A, \Theta)].$$

Quite unexpectedly, when taking $s(\cdot)$ as Bayesian surprise, using techniques from information geometry, the result is a flipped version of Shannon’s capacity-cost function (Shannon 1948; 1959; Varshney 2008).

**Theorem 1** (Varshney (2019)). The fundamental tradeoff between novelty and quality in combinatorial creativity is given by the following expression:

$$S(Q) = \max_{P_A(\alpha)} I(A, \Theta).$$  \hspace{1cm} (1)

Intriguingly, creativity has an equivalence to information transmission, with quality playing the role of energy and novelty playing the role of information rate. All of this, however, without communicative intent from the creator.
Introducing Intentionality

Now we extend the formalism to bring intentionality into the picture. In particular, we formalize intentionality as the need to reliably communicate a message \( m \) from the creator to an audience member using creative artifacts \( \alpha \), where perception of the message-bearing part of the creative artifact (the signal) is modeled as a noisy channel with transition probability assignment \( p_{\hat{A}|A} \). Here \( \hat{\alpha} \) are perceived signals, decoded as messages \( \hat{m} \). Now the goal is to communicate so error probability, \( Pr[m \neq \hat{m}] \), via creative artifacts is arbitrarily small.

For reliable communication alone, the limiting information rate, \( R \), is the channel capacity \( C \). If there are constraints on the signaling strategy, this may be reduced. Due to the noisy channel coding theorem, this fundamental limit of channel capacity is given as follows.

**Theorem 2** (Shannon (1948)). The fundamental limit of reliable communication in the presence of noise under the input constraint requiring the input distribution to be in the family \( \mathcal{P} \) is the channel capacity

\[
C(\mathcal{P}) = \max_{P(\alpha) \in \mathcal{P}} I(\mathcal{A};\hat{\mathcal{A}}). \tag{2}
\]

Now we essentially combine Theorem 1 (information geometry argument) with Theorem 2 (random coding argument) to get a limit theorem for creativity with intentionality. The detailed proof is omitted for brevity; a sketch is given.

**Theorem 3.** For a given perception channel \( p_{\hat{A}|A} \) and known set \( \Theta \), we require a minimal amount of average quality \( Q \) and minimal novelty \( S \), then the maximum information rate of communicative intent \( R \) that can be reliably transmitted is:

\[
C(Q,S) = \max_{P(\alpha) : \mathbb{E}[q(A)] \geq Q, I(\mathcal{A},\Theta) \geq S} I(\hat{\mathcal{A}};\mathcal{A}).
\]

**Proof.** Notice that Eq. (2) in Theorem 2 holds for any constrained family of input distributions \( \mathcal{P} \). Here we choose \( \mathcal{P} = \{ P(A) : \mathbb{E}[q(A)] \geq Q, I(\mathcal{A},\Theta) \geq S \} \) to satisfy the novelty and quality constraints and the result follows. Detailed arguments are needed to ensure that asymptotic arguments have appropriate interaction.

Again details omitted for brevity, but the information-theoretic optimization problem in Theorem 3 can be reformulated in a dual formulation as an optimization of novelty under quality and communicative intent constraints as:

\[
S(Q,R) = \max_{P(\alpha) : \mathbb{E}[q(A)] \geq Q, I(\mathcal{A},\Theta) \geq R} I(\mathcal{A},\Theta).
\]

to become more along the lines of the form of Eq. (1) in Theorem 1, which is a constrained optimization for novelty. As we can see in this form, the requirement for positive communicative intent rate may decrease the novelty that can be achieved, when the mutual information constraint is active.

Note that there is a natural Markov relationship as \( \hat{\mathcal{A}} \leftrightarrow \mathcal{A} \leftrightarrow \Theta \), based on the order of how variates are chosen. Hence, the constraint will indeed often be active.

In a third alternative form as a constrained optimization for quality, one can likewise observe that the communicative intent requirement may reduce the achievable quality.

As far as we know, this is the first characterization of how introducing intentionality into combinatorial creativity may reduce achievable novelty and quality.

With this limit expression in hand, we can also notice a formal connection to the information bottleneck functional (Tishby, Pereira, and Bialek 1999; Gilad-Bachrach, Navot, and Tishby 2003) which has become prevalent in the machine learning community and can be thought of as expressing the idea of minimal sufficient statistics with respect to a relevance variable. In particular, both settings are mutual information optimization under mutual information constraints but with the inequality going the other way. For creativity, one term is concerned with communication and the other with novelty, whereas for sufficient statistics, one term is still communication but the other is relevance. In this sense, it is interesting to think about novelty as a kind of irrelevance to the inspiration set (irrelevance rather than relevance due to the reversed inequality).

**Conclusion**

In this paper, we have summarized our recent work in engineering systems theory that uses information geometry to establish the fundamental limits of creativity and that may therefore be of interest to researchers in computational creativity. Importantly, (Varshney 2019a) had dismissed the role of intentionality in creativity when establishing limit theorems for the technical problem of creativity.

Here we have therefore brought intentionality back into the picture and established a further limit theorem for intentional creativity, connecting our previous limits of creativity with Shannon’s previous limits of reliable communication. This investigation of semantic creativity shows that requiring communicative intent may reduce the quality and/or novelty of creative artifacts that are generated. Moreover, connections to the information bottleneck in machine learning essentially show that novelty in creativity can be thought of as a kind of irrelevance to the inspiration set.

Going forward, it will be of interest to compute fundamental limits with and without intentionality in given creative domains, and to compare them with the performance of existing computational creativity algorithms.

**Acknowledgment**

Discussions with D. Oppenheim are appreciated.

**References**


A Deep Dive Into Exploring the Preference Hypervolume

Alexander Hagg1,2, Alexander Asteroth1, Thomas Bäck2
1Bonn-Rhein-Sieg University of Applied Sciences, 2Leiden Institute of Advanced Computer Science
Germany, The Netherlands
alexander.hagg@h-brs.de, alexander.asteroth@h-brs.de, t.h.baekc@liacs.leidenuniv.nl

Abstract

Computers can help us to trigger our intuition about how to solve a problem. But how does a computer take into account what a user wants and update these triggers? User preferences are hard to model as they are by nature vague, depend on the user’s background and are not always deterministic, changing depending on the context and process under which they were established. We pose that the process of preference discovery should be the object of interest in computer aided design or ideation. The process should be transparent, informative, interactive and intuitive. We formulate Hyper-Pref, a cyclic co-creative process between human and computer, which triggers the user’s intuition about what is possible and is updated according to what the user wants based on their decisions. We combine quality diversity algorithms, a divergent optimization method that can produce many, diverse solutions, with variational autoencoders to both model that diversity as well as the user’s preferences, discovering the preference hypervolume within large search spaces.

Introduction

Although we will never be able to fully and concisely grasp it, Hegel describes that in its core, the creative process strives to discover one’s true self. Art has the task to reflect upon the spectator and is necessarily interactive (Hegel 1842).

We experience intuitions about what we want but are often not able to formalize our preferences. They are based upon direct experience, on cross-connections we make based upon unrelated experiences, on familiarity but also on our problem solving skills (Raidl and Lubart 2001). We experience a physical sensation as a reaction to these intuitions, as defined by C.G. Jung, or more precisely, introverted intuitions, which are paramount to discovering one’s own preferences (Jung 1923). Research that provided evidence that intuition and creativity are positively related (Raidl and Lubart 2001) however shows that the debate on creativity and intuition has not been and might never be settled.

We use the definition of intuition as a sensation that arises when a perceived pattern is unconsciously matched to another formerly perceived one (Rosenblatt and Thickstun 1994). One source that can act as a stimulus to intuition is external (Raidl and Lubart 2001). The artist’s or creative engineer’s vigorous search cannot be performed in a vacuum. We might have an initial idea and the ability to perform a divergent search, yet need others to reflect upon themselves and gain true insight into what they are and want. Sartre emphasizes that the Other is needed for reflection, but the artist, every creator, or indeed everyone, has the responsibility to choose (Sartre and Elkaïm-Sartre 1946). In a creative process, especially when reflecting with others, we might use the Jungian ability of extraverted intuition, or brainstorming, to come up with and reflect upon novel solutions that do not coincide with our own intuition (Jung 1923).

In order to communicate and reflect upon our preferences with ourselves as with others we need to create examples that try to capture abstract ideas. Creativity, which Guilford defined as the ability to perform divergent thinking, is about generating many examples that adhere to one’s preferences (Guilford 1967). It is through these examples that we can both explore and communicate our preferences.

We combine the ideas of Guilford about divergent thinking, Jung on intuition and Sartre on reflection by others and pose that the central object of preference discovery in computational creativity should be the creative process that includes all three aspects, whereby the Other is represented by a computer and by other creators. We present an example of such a process, akin to the generative-exploratory creative model (Ward, Smith, and Finke 1999), using quality diversity algorithms to perform divergent search to trigger the intuition of the users, allowing them to take influence interactively. The computer feeds and reflects upon this interaction and merges the intuitions of what is possible and preferred in a model. The model and the quality diversity algorithm are the motor behind the creative process, in which computer and users co-create a common understanding.

Related Work

Evolutionary computation has often given us unexpected solutions to engineering problems (Lehman, Clune, and Mišević 2018). Novelty search (Lehman and Stanley 2011) took the idea of divergent search to a new level by abandoning the objective function altogether, its only goal to find a set of novel solutions. Reintroducing the objective to this purely divergent search method gave way to quality diversity (QD) algorithms like MAP-Elites (Cully et al. 2015). As in multimodal optimization (Preuss 2015), it finds a diverse set of high quality optimizers, but instead of performing niching in the search space directly, it does so in phenotypic...
or behavioral space. First applied to robotics, QD finds a large number of high-performing robot controller morphologies by only comparing fitness between similar solutions, in terms of their morphology or behavior. QD keeps track of solutions in an archive of niches and finds a subset of regions in genetic space, called the elite hypervolume (Vassiliades and Mouret 2018), or prototypes (Hagg, Asteroth, and Bäck 2018). Similarly, we can describe the volume that contains the preferred solutions the preference hypervolume.

A computer aided ideation process using QD can be developed that is based on an a posteriori articulation of preference (Hagg, Asteroth, and Bäck 2018), or “design by shopping” (Balling 1999). By using a preference model based on genetic similarity to selected solutions, and incorporating a factor in the objective function that rewards solutions that are closer to the selected ones, new solutions generated by the system are similar to the user’s selection (Hagg, Asteroth, and Bäck 2019). This approach depends on whether genomes that are closer together also are close in their expressed form, which cannot always be guaranteed. When a user prefers a solution, they would certainly expect the updated solutions to be similar in terms of their expressed morphology or behavior, not their encoding.

In reinforcement learning, learning a neural network model from human preferences makes it possible to find robust controllers without an explicit objective and instead showing the user pairs of examples and letting them pick which one they prefer (Christiano et al. 2017). Showing the user a rich set of high performing designs and having the user select designs is not new (Stump et al. 2003). Recent work shows that using generative or latent models that are trained on a diverse set of solutions (Fernandes, Correia, and Machado 2020) allow the user to easily search in the latent space created by the model. This allows interactive evolution of the latent model, for example to interactively recreate images (Bontrager et al. 2018). However, the latent space is per definition an interpolative space which seems to be less suited for ideation processes than an extrapolative space.

**HyperPref**

We introduce HyperPref, an implementation of the idea of integrating divergent thinking, intuition and reflection into an interactive co-creative process (see Fig. 1). The central process consists of two alternating steps: I) the computer initiates the process by producing a diverse set of high quality solutions and II) the users select the solutions they prefer, after which the computer updates the set of solutions that are preferred as well as high performing.

An initial pool of random solutions is generated and evaluated using a user-defined objective, which can be an optimality criterion or a more general criterion about the appearance of solutions, throwing a wider net for more “free” thinking. The genomes are then expressed into their phenotypes. A latent model, a variational autoencoder (VAE), is trained to compress the phenotypes into a low-dimensional description. This allows us to determine how similar solutions are and perform phenotypic niching despite of the high dimensionality of the phenotypes. QD creates such a latent niching archive, consisting of high-performing solutions (according to the user-defined objective function) and triggers a first intuition of what good solutions can look like.

![Figure 1: Discovering the preference hypervolume with HyperPref.](image)

Triggering their intuition, users select their preferred solutions from the archive, and a snapshot of the latent model and preferences is saved. By using a similarity metric based on the latent distance of new candidate solutions to the preferred and non-preferred solutions we can determine whether a new candidate solution is likely to be part of the preference hypervolume or not. The preferred solutions are used to create a new set of initial solutions by perturbing the original solutions and adding them to the set of originally selected ones, adding possibly new innovations into the data set. The preference hypervolume usually consists of disconnected regions in the search space. By increasing the mutation strength (the $\sigma$ of the normal distribution from which the amount of perturbation is chosen), we allow finding shapes between phenotypic clusters.

Note that in contrast to other work, we do not directly use the latent space for the search, only to compare phenotypic similarity of solutions. Although the VAE would allow this, it would constrain the search to the interpolated space between selected solutions. This would only be really sensible when the first latent model from which the users select solutions was trained on all feasible and relatively high performing solutions. Although QD is a strong mechanism to find diverse solutions, it does not guarantee that all solutions are found. We can also assume that the latent model will not be able to model all variations within the solution set,
especially not with low-dimensional latent space, which are necessary for QD to remain feasible. In conclusion, not limiting the search to the latent space will allow more innovative solutions to be found once the users constrain the search to their preferences, considering solutions that would not be considered by the first model.

The computer updates the latent model, which now describes the similarities within the preference hypervolume. By adjusting the objective function, adding the constraint model, the intuition is updated. This process can continue until the users are satisfied. Novelty search (as opposed to QD) and autoencoders have been combined (Liapis, Yannakakis, and Togelius 2013), but without involving an explicit external objective. Our generator searches for quality as well, and we use the autoencoder not as a way to enhance novelty but rather to capture the user’s choice.

Demonstration

HyperPref is demonstrated on a 2D shape domain, consisting of local interpolating splines (Catmull-Rom 1974). The splines are encoded by a polar coordinate based genome (see Fig. 2). By controlling the radius $r$ and angle $\theta$, a large variety of convex and concave shapes can be created.

We simulate two use cases. In an artistic case, the users are looking to design a ninja star, starting from centrally symmetric shapes. The second case, which is closer to creative engineering, starts out with unbalanced shapes with the objective to find wing profiles. The first objective prefers solutions that are point symmetric through the center point of the shape. The shape is sampled at $n = 100$ equidistant locations on its circumference, after which the symmetry metric is calculated. The metric is based on the symmetry error $E_s$, the sum of Euclidean distances of all $n/2$ opposing sampling locations to the center: $f_P(x) = \frac{1}{1 + E_s(x)}$, $E_s(x) = \sum_{j=1}^{n/2} \| x_j - x_{j+n/2} \|$.

The second objective maximizes the distance between the center of mass and the center of the bounding box around the shape.

For simplicity, we use a 2D latent space which only captures the similarity of solutions based on the largest phenotypic variance. The shape genome consists of 16 genes. QD produces 32 new child solutions for 1024 generations, by using a normally distributed mutation operator with $\sigma = 10\%$ of the parameters’ range as a generator of diversity. The archive holds $20 \times 20$ solutions. The convolutional VAE is trained on a GPU with 128 by 128 pixel representations of the shapes and the ADAM (Kingma and Ba 2014) training method. The encoder consists of two convolutional and nonlinear (ReLU) layers, eight filters of size three with a stride of one. Training is performed with a learning rate of 0.001 and maximizes the evidence lower bound, thereby minimizing the Kullback-Leibler divergence between the original and the latent distribution. The solution set is updated using as many perturbed ($\sigma = 10\%$) versions of the selected shapes as the 64 initial solutions. The constraint penalty, which is multiplied with the original fitness function, is based on the user selection drift (Hagg, Asteroth, and Bäck 2019), with the minimal distance $s$ of a candidate solution $x$ to a selected solution and $\tilde{s}$ to a deselected solution:

$$p(x) = \begin{cases} 1, & \text{if } \frac{\tilde{s}}{s} < 0.5 \\ 1 - 2 \cdot \left( \frac{\tilde{s}}{s} - 0.5 \right), & \text{otherwise} \end{cases}$$

Experiments

Fig. 3 shows the initial computer-generated solution set on the left. The diversity of the sets is clearly visible. We then simulate a group of creators that all have a different preferred shape (shown in the center). After selection, the computer updates the set, reflecting the combination of the creators’ choice and the general objective. This process of user selection is repeated once more. The resulting sets of ninja stars or wing profiles to look like is shown in the second preference hypervolume on the right.

Discussion

The primary solution set contains a large and diverse number of shapes, offering users inspiration and feeding their intuition about how shapes could look like. With a specific goal in mind, namely designing ninja stars or wing profiles, users and computers can co-create in an intuitive creative process. The process offers reflection, by combining preferred shapes and zooming in on the preference hypervolume. Only two steps are necessary to create shapes that are close to what one could and would expect from such a creative process.
Conclusion

We showed how to combine the divergent search of quality diversity to trigger the user intuition about what solutions are possible and high performing, allowing creators to select shapes they prefer by shopping for designs, and then having the computer reflect upon that selection, incorporating the preferences through a constraint model and discovering the preference hypervolume in an intuitive, co-creative manner. The features upon which the initial set is based are generated by a variational autoencoder, trained on the phenotypical expression of the solutions, rather than hand-crafted features or genetic similarity. The constraint model is based upon a snapshot of that model in combination with the set of selected and non-selected solutions.

The resulting creative process, which continuously visualizes and updates the creators’ intuition, was shown in a simple 2D shape domain. The updates can be fast, depending on the GPU used for training the VAE and the number of QD updates and VAE prediction speed. The current bandwidth of GPUs is such that the method is close to being on-line.

We recognize that optimizations and variations of the introduced process exist. We used a 2D latent space for the purposes of simplicity and visualization, but higher-dimensional latent spaces are more accurate in measuring similarity in detailed shapes. An interesting research path will be to analyze the differences between searching the VAEs latent space, interpolating between selected shapes, and searching genetic space directly, allowing extrapolation away from the modeled surface, which seems to be more fitting for a creative process. Often times we only find innovative solutions during the creative process, and we certainly hope that unexpected, novel solutions are discovered once we made the first few design decisions.

We took a short but deep dive into exploring the preference hypervolume, combining quality diversity with latent models and interactive user selection into a co-creative process that shows what we can expect in the near future when creators work together. We put the human into the loop by feeding and reflecting upon their intuitions, leading the creative process by example: *No one can tell what the painting of tomorrow will be like; one cannot judge a painting until it is done* (Sartre and Elkaïm-Sartre 1946).

References


Jung, C. 1923. Psychological types: or the psychology of individuation.


Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2013. Designer modeling for personalized game content creation tools. In Ninth Artificial Intelligence and Interactive Digital Entertainment Conference.


The societal and ethical relevance of computational creativity

Michele Loi
University of Malta
michele.loi@ibme.uzh.ch

Eleonora Viganó
University of Malta
eleonora.vigano@ibme.uzh.ch

Lonneke van der Plas
University of Malta
lonneke.vanderplas@um.edu.mt

Abstract

In this paper, we provide a philosophical account of the value of creative systems for individuals and society. We characterize creativity in very broad philosophical terms, encompassing natural, existential, and social creative processes, such as natural evolution and entrepreneurship, and explain why creativity understood in this way is instrumental for advancing human well-being in the long term. We then explain why current mainstream AI tends to be anti-creative, which means that there are moral costs of employing this type of AI in human endeavors, although computational systems that involve creativity are on the rise. In conclusion, there is an argument for ethics to be more hospitable to creativity-enabling AI, which can also be in a trade-off with other values promoted in AI ethics, such as its explainability and accuracy.

Introduction

Creativity is beneficial to many aspects of human life, from the natural to the individual, to the social level. Current mainstream AI systems risk hindering processes that enable creativity and thus risk to reduce their benefits. In this contribution, we argue for the ethical and social benefits of creativity, identify the potential threats of AI for creativity, and present AI systems that enable creativity. The scope of our argument concerns both creative AI (i.e. AI generating valuable novelty directly) and creativity-enabling AI (i.e. AI that enables or favors the generation of valuable novelty by humans). We reason that by creating a bridge between the fields of AI ethics/philosophy and computational creativity (CC), and highlighting the threats of mainstream AI on human well-being, the societal uptake of CC can be promoted and lead to focal research in the field.

Most philosophical definitions of creativity involve the creation of something novel (Gaut 2010) i.e. originality. Something can be new in the sense of it being the first time that it has been produced in history (H-creativity) or in a person’s life (P-creativity) (Boden 1991). The dominant traditions see creativity as essentially related to the production of something valuable (Gaut 2010), i.e. creativity is essentially not just novelty, but valuable novelty. We shall not assume that a process condition - e.g. a special form of independence from past models (Kronfeldner 2009) - defines what creativity is. On the contrary, we shall show that certain processes just happen to favor creativity, i.e. favor or enable the production of valuable novelty. Thus the attribution of creativity to certain processes is based on observations and does not result from a choice of definition.

Processes enabling creativity

Natural creativity. The natural evolution of living entities fits our definition of a creative process. This view has been influential for attempts to model creativity computationally (Campbell 1960) as evolution, clearly, involves the production of novelty (H creativity) (Boden 2018), which is valuable for the organism (i.e. adaptive) or as a means to humankind. This kind of creativity is Darwinian, in that it is based on blind variation and subsequent selection. “Blind” genetic variation may not be “random”, but it is at least “undirected” (Boden 2018).

Social creativity. The collective dimension of creativity is enabled by liberty, both social and economic liberty. Social liberty allows for individual and collective “experiments of life” (Mill 1859) and for discovering new forms of social value previously thought impossible, e.g. stable relationship grounded in homosexual love and sexual conduct. The different forms of the good, in Mill’s view, cannot be discovered by pure intellectual acts of understanding, but must be lived out concretely (Anderson 1991). Creativity is also expressed by markets (the creative destruction of capital) (Schumpeter 1965). Markets resemble natural systems, even if product innovations are not random or undirected, but on the whole they are rather bets on future success. Since most entrepreneurs do not know whether their enterprise will succeed, the market is (in the short term) a highly inefficient system: predicting company success is highly inaccurate and 90% of startups fail. Yet, in order to produce the novelties that we highly value, “[n]atural and nature like systems want some overconfidence on the part of individual economic agents, […] provided that their failure does not impact others […]”(Taleb 2012). Individual entrepreneurial ignorance works like a high-risk bet, which enables exploration (i.e. of unknown and unpredictable consumer preferences) and is ultimately responsible for generating valuable novelty for consumers.

Individual existential creativity. In the individual dimension, creativity requires the individual’s attitude of exploring life possibilities and experimenting life plans and versions of herself. In this process, the individual deepens the knowl-
edge of herself, acquires life experience, and tests the life track that she previously chose. Individual existential creativity is thus not exclusively the purview of the artist, but of any individual that adopts the exploration of life opportunities and her potential as a pivotal value in her life. Individual existential creativity refers to how the individual is leading her life; therefore, it can be assessed with the standards of the philosophical conception of prudence. While, in the common meaning, prudence is conceived as a cautious attitude toward risk and danger, in philosophy, it is the pursuit of one’s own good throughout a lifetime. Individual existential creativity contributes to “philosophical prudence” as it is a way to live one’s life and pursue one’s good. One fundamental attitude that enables creativity is the openness to the unbidden, which is a disposition of recognition and acceptance that not everything is, or should be, under one’s control (Sandel 2007). The openness to the unbidden favors the acceptance of a world sometimes characterized by extreme variability and a mental disposition in which not everything has to be planned in detail. For example, the inclination of a parent to accept his or her child, however unexpected and different the child ends up being from the parent, accommodates biological variance. Creativity in life also involves the individual’s experimentation with different possible selves, namely trying various life tracks, each realizing a different idea of herself. Typically, in our culture, the experimentation phase has been confined to adolescence and the consolidation and exploitation phase to all later life stages. However, it is likely that in the future, with an increasing acceleration of social change and technology, life-plans will be built in such a way that experimentation phases can also occur later in life. This will require society to both encourage openness and experimentation during people’s lifespans and governments to provide a safety net to help them start again, in case of failure.

What these processes have in common

Summarizing the cross-domain analysis so far, we can list the following features of processes enabling creativity:

1. some degree of randomness, or at least, undirected processes, or at the very least non-fully-rationally-directed processes (e.g. processes directed in ways that arerationally defective or that lack rationality altogether). At the natural level, we have random mutations; at the social level, we have free initiative in markets and creative pursuits; and at the individual level, we have experimentation with a set of values and commitments that do not have a guaranteed payoff.

2. the generation of more variance than in the case of non-random, directed, and fully rational processes. For example, new life forms emerge through mutations, new needs are discovered when new products and services are invented, and new forms of life and values are discovered through social experimentation.

3. the generation of costs, that is, additional costs compared to non-random, directed, and fully rational processes. These processes are wasteful, and can even generate harm (e.g. harmful mutation). At the natural level, life evolves

in spite (and because of) many failed experiments. At the social level, resources are wasted, for example by failed businesses. At the individual level, the costs of creativity lay in the inefficient pursuit of a plan, commitment, career, etc. In fact, when the individual pursues a high-risk plan, she may end up with less satisfaction, happiness, and resources than with a low risk-plan, and the costs of recovery may be high.

4. limited reach of the damage, and “part-is-sacrificed-for-the-whole” principle. Most mutations are harmful but biological processes are flexible, most start-ups fail but consumers benefit, some experiments of life may be harmful to individuals, but societies learn that new forms of the human good are possible. At the individual level, while some life experiments pursued in a specific phase of the individual’s life may fail, subsequent experiments would typically benefit from self-knowledge and knowledge of reality gained with earlier unsuccessful trials.

Of these four features, (3) is positively harmful, (4) is beneficial, (1) and (2) are valuable as means. Our hypothesis is that both non-directedness and variance are means to exploring the space of possibilities (and generating knowledge), in particular, possible biological forms, preferences and needs, and shapes of the good human life. At the individual level, what gets explored is the individual’s potential and how to realize it. This is crucial for self-knowledge and authenticity, which are arguably essential to well-being (Sumner 1996).

In conclusion, creativity is the production of valuable novelty, which contributes to generating knowledge. The latter is especially valuable for humans as it enables them to adapt to the rather unpredictable world outside them and discover who they, collectively and individually, are.
The threat of current mainstream AI systems

Collective creativity is threatened by AI technologies that currently prevail. Let us take economic liberty as an example. We explained how the individual’s ignorance regarding the utility function and the willingness to take risks are important for the system to create novelty. A society in which decision making is highly informed by intelligent systems that are trained in a supervised fashion with a narrowly defined utility function that tries to avoid risk will not exhibit the same exploration power as a system based on the individuals’ judgments. Moreover, in this new, efficiency-driven reality, fewer agents will be taking over the decision making that was previously done by many more individuals. This leads to uniformity in the decision making process, which will lead to less diversity in the collective “experiments of life” we referred to earlier. However, these are crucial for creative processes in society. Lastly, we see an additional threat in the training cycle of these systems. They learn from past data labelled with relevant outcomes. The more intelligent systems come to influence decision making in society, the more they will impoverish their own training data, as the future data that serves to train them will over time become increasingly compatible with the original system expectations, since the data itself is influenced by these systems. The data will be more and more uniform, leaving many possible regions of the search space unexplored.

At the individual level, many decisions affecting the individual’s opportunities and plans are implemented by systems based on AI such as recommendation systems and digital wellness technologies. These systems influence the individual’s choice in various ways, by means of the visual presentation of the alternative as well as by the alternatives that are suggested by the system. These systems have two features that result in identity stagnation. First, current AI technologies maximize the present self’s utility function and thus present the individual with choice alternatives that satisfy her current preferences. This implies that such technologies make identity and preference changes less likely. Second, the profiling of the future self provided by current AI systems is based on past data (Mittelstadt et al. 2016). The latter excludes unlikely but disruptive events from the predictions of the algorithms; therefore, current AI systems nudge the user to continue on the current life track. This identity stagnation decreases the individual’s overall well-being by limiting the possibility to experiment with new life tracks and learn from them, and make the attitude of openness to the unbidden useless, as the user is never provided with unexpected options.

Computational systems involving creativity

However, not all work in AI is harmful for creative processes, on the contrary, and the ICCC conference series are the living proof of that. Although for this short paper, an exhaustive survey is out of scope, we would like to give a short (and certainly incomplete) overview of computational systems that involve aspects of creativity. To organize this wide spectrum of systems, we can determine what aspect of creativity they are focusing on, the creative product or the creative process, which are two main aspects of study (Said-Metwaly (2017), among others). Computational work that focuses on the first aspect includes work on the generation of unconventional linguistic variations, such as the generation of metaphors (Veale and Hao 2008), scientific ideas (Wang et al. 2019), or visual art (Elgammal et al. 2017). Although automatically generating creative output requires putting thought into the creative process, the main focus of these works is on the creative output.

We would like to zoom in on computational work, where the focus is on the creative process. Some work builds computational models to better understand the cognitive process of creativity. Works from the field of cognitive science, for example, have shown evidence that creative people have more complex semantic network structure and may activate a wider range of associations across the network than less creative people (Kenett, Anaki, and Faust 2014).

Other work focuses on creating computational algorithms that introduce aspects of novelty in the learning process. Many are from the field of robotics, for example, they describe situations in which robots need to navigate in an unknown environment and need to look for novelty in order to better explore the space and not get stuck in local optima. One type of solutions solves this task by defining a so-called intrinsic reward (based on the psychological concept of curiosity (Barto (2013), among others). Intrinsically Motivated Reinforcement Learning (Kaplan and Oudeyer (2007) among others) basically works as follows: the agent is rewarded for discovering new patterns in the environment (Schmidhuber 2010), and it is always in search of novelty. Other work aims at modelling social creativity specifically (Saunders 2019). The author reports on experiments, in which multiple agents with diverse hedonic functions, which determine levels of interest and actions to be taken, work together. Another strand of research introduce aspects of novelty in the learning process under the header of evolutionary computing (EC). An approach known as Novelty Search (Lehman and Stanley 2011) searches for behavioral novelty instead of seeking an objective. Also here, we see EC algorithms that are built to run in a distributed fashion over a population of agents and works in which evolution occurs within one agent only, usually employing a time-sharing strategy of genes.

In summary, we find several algorithms that involve aspects of creativity. These algorithms exhibit the four characteristics of creative processes we listed previously: aspects of undirectedness, generation of variance, generation of costs in order to increase knowledge, and the part-is-sacrificed-for-the-whole principle. They also implement these both at the individual level of single agents and at the level of social interaction. However, these algorithms are not the mainstream algorithms that find their way to the market, where narrowly defined objective-driven high-accuracy systems prevail. Although even here we recently see some changes taking place. Recommender systems, such as video recommendation systems for Youtube, incorporate reinforcement learning, and off-policy learning to avoid ‘myopic recommendations’, where the short term reward overshadows long-term user utility in the form of dis-
covery content (Ma et al. 2020). Motivations here come from trying to optimize the long-term user utility. We feel that attention to the ethical aspects, related to the threats the current AI technologies pose, could spearhead the work on computational creativity.

Conclusion

Individual and collective creativity are ethically valuable because they are essential to (a) adaptation to the unexpected; (b) self-knowledge. Both adaptation and self-knowledge are "permanent interests of man as a progressive being" (Mill 1859). Adaptation is a permanent interest of humanity because humans face environments that are unpredictable in the long term. Self-knowledge is essential to well-being because a life cannot be good for an individual unless it reflects her individuality, and individual preferences must be informed (including, by self-knowledge) in order for happiness to be authentic (Sunner 1996).

Current AI ethics guideline documents (Jobin, Ienca, and Vayena 2019) mention freedom and autonomy as higher-order principles. But in those guidelines the focus is on protecting human freedom and autonomy, typically against the overreach of poorly controlled AIs. When freedom and autonomy are addressed, there is no reflection on the idea that the autonomy and creativity of AIs may also be needed as enablers of human freedom, autonomy, and well-being. In our analysis, computational creativity may be needed for human well-being and it is thus in a trade off with other legitimate goals of AI (e.g. accuracy). We hope that this brief reflection motivates scholars of computational creativity to reflect on the ethical importance of their discipline and contribute to AI ethics by researching open questions such as, how to build systems that integrate computational creativity with resilience and how to avoid excessive harm.

Acknowledgments

This work was partly funded by the Digital Society Initiative of the University of Zurich and the Cogito Foundation grant 17-117-S. We would like to thank the reviewers, and Paul O. Dehaye and Albert Gatt for their valuable suggestions.

References

Can a Robot Do a Trust Fall?
Absurdity as a Component of Human Intelligence and Embodiment
Amy LaViers and Ilya Vidrin

Department of Mechanical Science and Engineering and Department of Theatre
University of Illinois at Urbana-Champaign and Northeastern University
Urbana, IL 61801 USA and Boston, MA 02115 USA
alaviers@illinois.edu and i.vidrin@northeastern.edu

Abstract

Trust is often considered valuable in a broad range of relationships, from professional collaborations to personal partnerships. This article examines the possibility of trust in a robotic system. By posing the question “can a robot do a trust fall?”, an investigation on the issues embedded in designing trusting systems is presented, using methods and perspectives from philosophy and engineering. Posing such a question helps us understand the physicality and embodiment of trust, as well as the limits and resources of robotics.

Introduction

There have been a number of recent moves in the philosophy of emerging media to distinguish between the concepts of “reckoning” and “judgment”. The distinction holds that “reckoning” is a limited form of “calculative rationality”, while judgment is linked to full-blooded intelligence “that is existentially committed to its own existence and to the integrity of the world as world” (Smith 2019). The distinction between reckoning and judgment is particularly useful for differentiating what systems are capable of, including their limitations and resources.

One overlapping aspect of judgment and reckoning seems to be a criteria of rationality. Yet humans can and do behave irrationally. In fact, acting irrationally is a natural part of human experience. Moreover, we are able to learn and grow from our own irrationality (Ashcroft, Childs, and Myers 2016). Likewise, seemingly integral to human understanding and experience is the phenomenon of absurdity.

One example of absurd behavior is risk-taking. While taking “calculated risks” are a necessary component of common-place actions such as signing automotive or home loans and investing in higher education, there are more free-form, aesthetic risks that are just as ubiquitous. Bungee jumping and riding roller-coasters are two examples of risky behaviors that are motivated by aesthetic ends.

When we undertake a risk, we may trust that everything will be fine. When riding a roller coaster, for example, an individual can fully believe that the ride is not actually risky, thus placing a certain limited trust in the machinery. We may also not care about the results of our risky endeavors, such as if an individual does not care if things turn out poorly (in which case it does not matter what they believe or trust). Yet, without a genuine feeling of risk, the roller coaster has little aesthetic value to the rider.

Returning to judgment and reckoning, people who engage in risky behavior need not weigh options in a deliberative, “rational” manner. They can (and do) engage in irrational behavior in order to pursue particular ends (e.g., ecstatic states) in the moment. It is perhaps even possible that individuals can occupy two seemingly opposite states simultaneously: it is raining, but I do not believe that it is raining. This statement is an example of Moore’s paradox (Moore 1993). While it is not something that we reasonably assert, we argue that there is still some value in such claims. After all, risky behavior is not necessarily rational.

Certain risky experiences can be relationship-building when undertaken together. For example, in a “trust fall” exercise, where one falls backwards into the arms of another, we can experience a moment of wild abandon. Rather than promoting our own individual homeostasis and livelihood, the exercise can serve as a ritual of putting trust in another person. Without the risk of falling to the ground, the exercise does not work, as we will outline further in this paper.

Perhaps this activity is an example of exemplification (Elgin 2010), wherein certain qualities of the experience of the action are non-propositional. Aesthetic ends that are manifest through creative, non-deliberative tasks may also require exemplification to advance understanding (Elgin 2010). Machines, governed by circuitry that implement first-order Boolean logic, are in some sense behaviorally limited by this. Thus, here, we propose a question: can a robot do a trust fall?

In this paper we will examine the conditions of executing a genuine trust fall, followed by an examination of why creating such a protocol for a robotic system fails. We conclude by posing questions around the potential insights of this investigation.

Defining a “Trust Fall”

There are many variations of the trust fall exercise. One typical version occurs within a dyad: while standing, one person, the “faller”, closes their eyes and falls backwards, relying on being caught by a “spotter” before hitting the ground. The exercise is based on the assumption that the act of reliance, when successfully executed, builds trust between the two individuals. A trust fall involves some kind of agreement on both sides – the faller agrees to fall and the spotter
agrees to catch. The full force of this agreement obligates the faller to be vulnerable and obligates the spotter to be receptive. If the faller does not care about being injured or is not afraid of hitting the ground, then the exercise will not function effectively to build trust between partners.

There are two paradigmatic cases worth exploring 1) the spotter does not succeed in catching the faller, and 2) the faller stumbles, flinches, or abandons the fall (consciously or not). In the first case, the spotter fails to uphold the obligation of catching. This is the potential fear of the faller. Even if the spotter is prepared and ready, the faller must allow themselves to be vulnerable. This can be quite an absurd task, in which one agrees to be vulnerable at the expense of not being caught or even getting injured. In the second case, the faller fails to deliberately fall. That is, in stumbling, bending their legs, or twisting around at the last moment to face forward, the faller “falls cautiously” and exemplifies a lack of trust in the catcher.

There are further problems inherent in the physical action. The exercise can become too “practiced”, wherein the faller is no longer being genuinely vulnerable. By extension, the exercise is no longer “abrupt”. To promote vulnerability, the spotter may change at which point they catch the faller, such that the faller does not know when they will be caught. But the faller may not care about being vulnerable, since even if the catcher is deliberately catching, the faller is not fully realizing the exercise of deliberately falling. Feelings of trust may thus be unidirectional, in that the faller trusts the catcher but the catcher does not trust the faller or vice versa.

We have an assumption that we are not designed to fall. As such, to fall is to place oneself at risk – a seemingly absurd action. The catcher is sensitive to the risk of the faller, just as the faller is sensitive to being caught. Thus it is not possible to establish genuine trust through physical interaction if the trust is unidirectional. This can be better illustrated through an example of catching a falling plate. Sensing that a plate is falling, I lean forward to catch it. Success in catching may lead to a sense of accomplishment or even augmented self-worth, but the action has not served to build trust between me and the plate.

We recognize that a “true” (as defined above) trust fall is a rarity. The exercise is contingent on skill and orientation of each agent toward the other, including such as factors as mood and environment. Thus a “true” trust fall cannot be arbitrarily created by any two humans on any given day. This excludes (the unlikely scenario) of tripping into a trust fall (versus deliberately falling). Notably, the component of trust features prominently in the initiation of falling on the part of the faller. Ostensibly, trust can only be built if the faller is genuinely vulnerable, meaning they are afraid of falling and choose to fall anyway. To design a system in this way means accounting for the way in which a system goes outside of its own protocol.

Formulating a Policy (Machine)

Moore’s paradox provides an example that we have argued demonstrates how absurdity can be key to human understanding and experience. This section provides an attempt to create a computational structure that encompasses the case wherein a system holds both the proposition and its counter in validity at the same moment. To format this attempt, we will use the general structure of a transition system, which can describe the architecture of software-controlled engineered systems. In general, we think of this transition system as creating a discretized policy for our machine.

To model the possible structure for the action of the faller (noting that the catcher has a similar structure that we will not explicate here), let us define our transition system as a tuple of a set of 1) $Q$, states, 2) $q_0$, initial states, 3) $E$, events or “actions” that can be taken from a given state, 4) $o$, an output function that associates state-event pairs to a new system state, 5) $\pi$ higher-level propositions that are true or false at each state, and 6) $h$, a labeling function that associates states to propositions.

This definition will be used to describe the machine’s current state and which states it may evolve to from that state, via which available actions. It provides a substrate onto which we can apply first-order predicate logic structure, enacting a series of propositions based on the structure of a formula $\phi$, expressed in linear temporal logic. Specifically, we consider temporal operators $\text{next} \ X$, and $\text{until} \ U$, and logical operators $\land$, $\lor$, and $\neg$, $\rightarrow$. Other logical operators like or and implication $\rightarrow$ can be expressed through combinations of these operators, i.e., disjunction is $\phi_1 \lor \phi_2 := \neg(\neg\phi_1 \land \neg\phi_2)$. Similarly, aggregated temporal concepts like eventually and always $G$ can be defined, i.e., $G\phi := \neg F\neg\phi$. The formula $G\phi$ states that proposition $\phi$ holds at all states of operation. Such a formula can be represented as a Büchi Automaton as in (LaViers et al. 2011). Changes in state and event structure can also be enacted to try and capture the phenomenon described in the previous section. We will use this abstraction to concretize our comments about trust through motion.

To create an initial model that may capture this desired behavior, assume that we have a sensing system in place that is able to detect gross physical states, i.e., by calculating and integrating the various measures of the machine sensors. These states are defined relative to the machine’s stability, e.g., with a static stability detector, we may be considering simply whether the center of mass is inside the polygon of support or not. Similarly, on our event structure, we have an actuation system that computes steps needed to either increase or decrease stability.1

Thus, the initial model of our system becomes:

$$T_1 = (Q_1, q_{1_0}, E_1, \Pi_1, h_1)$$

where

1) $Q_1 = \{\text{stable, unstable}\}$;
2) $q_{1_0} = \{\text{stable}\}$;
3) $E_1 = \{\text{stabilize, mobilize}\}$ offers two action options;
4) $o : Q_1 \times E_1 \rightarrow Q_1$, as shown in Fig. 1;
5) $\Pi_1 = \{\text{falling, not falling}\}$; and

1The concern of this paper is not on creating a state-of-the-art stability detector or generator, so we leave these details to the imagination of the reader.
6) \( h_1 : Q_1 \rightarrow 2^{\Pi_1} \) makes associations between stability and falling, e.g., \( h(\{\text{stable}\}) = \{\text{not falling}\} \) and \( h(\{\text{unstable}\}) = \{\text{falling}\} \).

![Figure 1: Visualization of \( T_1 \), showing state, event, and proposition structure.](image)

We could create a robot that simply always mobilizes in an unstable state – this is a robot that would always fall over. Instead, we aim to design a robot that associates the instability inherent to the state of falling with both propositions “falling” and “not falling”. We can use propositional logic to construct a controller for the system, creating a supervisory machine that will enforce the structure of these propositions as in (LaViers et al. 2011). For example, during a trust fall we could run GFalling and when not engaged in that activity we could run G¬falling.

This abstraction of a machine will “function” correctly – it will stabilize itself when in an unstable state and “fall” in a trust fall. However, in a “trust fall”, as discussed in the prior section, we want to create a state that can be pursued which is simultaneously “falling” and “not falling”. Thus, an additional proposition needs to be associated with the action of mobilization. In this case, the faller needs to mobilize with a **true** vulnerability that falling may occur and, at the same time, a **true** confidence that it will be caught.

Thus, an updated model of our system might become:

\[
T_2 = (Q_1, q_{1t}, E_1, \Pi_1, h_2)
\]

where

\[
h_2 : Q_1 \rightarrow 2^{\Pi_1}
\]

makes associations between stability and falling with the added nuance that in some unstable situations the machine may also be “trusting”, e.g., “not falling” as shown in Fig. 2.

![Figure 2: Visualization of \( T_2 \), showing state, event, and proposition structure.](image)

Now, we need a controller that executes the proposition \( G(\text{falling} \land \neg \text{not falling}) \) (during a trust fall) and \( G^\top \text{¬falling} \) (during a non-trust fall). We can see this fails because either the stable or unstable state satisfy “not falling”, resulting in a robot that may always fall. Another attempt may create distinct states and events associated with this activity, e.g.,

\[
T_3 = (Q_3, q_{1t}, E_3, \Pi_3, h_3)
\]

where \( Q_3 = \{\text{stable}, \text{unstable}, \text{unstable but safe}\} \);

\[
E_3 = \{\text{stabilize, mobilize, mobilize cautiously}\}
\]

\[
\Pi_3 = \{\text{falling, not falling, falling and not falling}\}
\]

\[
h_3 : Q_3 \rightarrow 2^{\Pi_3}
\]

makes associations between stability and falling with the added nuance that distinguishes unstable situations as shown in Fig. 3.

This structure would allow for a state designated as “falling and not falling” but it requires two distinct mobilization activities (to avoid being nondeterministic) and a new **unstable state**, meaning the machine is not in an **authentic** unstable state during the trust fall. Moreover, we must create a third proposition to label this state, as giving it both “falling” and “not falling” would result in the same error we encountered with \( T_2 \).

This example demonstrates a puzzling phenomenon for robotics and AI researchers to reconcile: curiosity seems to be an inherent and critical feature of natural intelligence. Yet, the act of being curious involves making mistakes, taking actions for “no reason”, and, more generally, engaging in and being attracted to “play”, activity that is inherently orthogonal to some other activity that may be defined as “work”. (We could, also, build a robot designed to, say, play with children, then the “work” of this robot is, indeed, play.) Thus, the characteristic of being curious involves taking actions that are inherently off policy, and there is a circularity that we seem to not be able to get out of: if we create a policy that guides the machine to go off policy, the machine is simply obeying a broader policy from which it cannot deviate. Thus, can a machine enter the state of absurdity that is necessary to complete a genuine trust fall? Can a machine build genuine trust with humans, who seemingly do take on these states of vulnerability, if it cannot inhabit an off-policy position?

**Discussion**

Full-blooded trust requires some kind of judgment. Since the medium of the fall is movement, however, the judgment need not necessarily be deliberative. That is, the faller intentionally falls, without needing to deliberate by weighing options about when or how to do so. We understand a deliberative judgment as one that requires a serial process of weighing options, in which either individual (or both) assess the risks involved of falling and catching. A non-deliberative judgment is exercised in the experiential receptivity to the moment of falling. In other words, non-deliberative judgment is keyed into the true state. This need not be considered outside of conscious awareness (whatever that might mean biologically). That is, a feature of what we take to be embodied reasoning may or may not be something we are consciously thinking about, but it is still a feature of intelligence we are interested in examining.

We note that absurdity, curiosity, and vulnerability seem to be features of human intelligence that are keyed into embodiment, and that these features are required for activities like building trust and being creative. The trust fall is an
exercise used in many community building and conflict mediation activities, but it is also a metaphor for many kinds of human relationship building activities, e.g., exchanging vows, sharing personal secrets, or appreciating ironic situations (e.g., rain on your wedding day). The formulation suggested here of a trust fall as an exercise requiring a moment of absurdity, vulnerability, and uncertainty from both agents (“Am I going to be caught?” and “Will I catch them?”) and of a machine’s policy as providing a forever level of reasonableness, resilience, and certainty, draw into question whether a machine can ever exhibit key features of human intelligence.

The trust fall seems to fall outside of first-order linear predicate logic. Specifying different qualities, textures, or even expectations means that trust is not just a product of a binary yes or no. Perhaps then, a trust fall establishes a bounded, three-part predicate of trust, such that the faller can reasonably say: I trust you to catch me if I fall. Even in successful circumstances in which the catcher has demonstrated their ability to catch the faller, it may be unreasonable for the faller to then assume complete trust in the catcher such that they say something like I trust you to save me from a burning building. Three-part predicates could also be further complicated by expressive qualifiers: I trust you to catch me gracefully or carefully. Exploring different logical structures may prove more fruitful to explain a genuine trust fall.

**Conclusion**

Modeling is a useful tool for gaining understanding of limitations, as much if not more than beneficial resources. The limits of modeling trust are located in the fact that trust has affective, expressive dimensions, but affective dimensions are not deterministic of mental attitudes. We argue that the affective dimension of trust makes it essential to experience rather than merely to measure it. Trust is not subject to a bounded binary of ‘yes or no’, but rather involves a richer set of behaviors. This means that accomplishing trust with computerized machines needs to account for the dynamic density of human experience, affording genuine, rather than simulated, reduced, or approximated, trust.

We argue that occupying absurd states helps us process ourselves and understand each other (the process of curiosity: truly going off policy and being uncertain). We have presented models of a trust fall in which $p$ and $\neg p$ is internally consistent and suggest that perhaps exhibiting this paradox is an important piece of intelligence. This may highlight something humans do in embodied form, which may be termed somatic intelligence, emotional intelligence, intrinsic motivation, etc.

The notion that absurdity – or this internal consistency that breaks logic – may be necessary to the formation of genuine trust has implications for human-machine dyads or teams – especially in the growing field of human-robot interaction where machines are treated as agents inside social contracts. If machines cannot replicate the state of falling and not falling, then perhaps any notion of “trust” in these relationships requires new structures, terminology, and paradigms to be accurately described.

**Acknowledgments**

The authors contributed equally to the writing of this manuscript.

**References**


Bridging Generative Deep Learning and Computational Creativity

Sebastian Berns$^1$ and Simon Colton$^{1,2}$

$^1$Game AI Group, EECS, Queen Mary University of London, United Kingdom
$^2$SensiLab, Faculty of IT, Monash University, Australia

Abstract

We aim to help bridge the research fields of generative deep learning and computational creativity by way of the creative AI community, and to advocate the common objective of more creatively autonomous generative learning systems. We argue here that generative deep learning models are inherently limited in their creative abilities because of a focus on learning for perfection. To highlight this, we present a series of techniques which actively diverge from standard usage of deep learning, with the specific intention of producing novel and interesting artefacts. We sketch out some avenues for improvement of the training and application of generative models and discuss how previous work on the evaluation of novelty in a computational creativity setting could be harnessed for such improvements. We end by describing how a two-way bridge between the research fields could be built.

Introduction

Methods in generative deep learning have become very good at producing high quality artefacts with much cultural value. In particular, Generative Adversarial Networks (GANs) (Goodfellow et al. 2014) provide many exciting opportunities for image generation. Over the course of only a few years, research contributions have pushed models from generating crude low-res images to ones evoking visual indeterminacy, i.e., images which “appear to depict real scenes, but, on closer examination, defy coherent spatial interpretation” (Hertzmann 2019) and even further, to generate images of human faces indistinguishable from digital photographs (Karras, Laine, and Aila 2019). These are just a few of the many applications of generative deep learning around which the notion of ‘creative AI’ has emerged.

Closely following the fast-paced research on neural networks and generative models in particular, an online community has formed under the hashtag #CreativeAI (Cook and Colton 2018), that has been particularly eager and successful in exploring unconventional applications and has established its place in workshops at major conferences with a focus on artificial neural networks. With extensive knowledge and experience in the development, application and evaluation of machine creativity, the computational creativity community can contribute to this progress by laying out potential ways forward. We aim here to build a bridge between generative deep learning and computational creativity by way of the creative AI community, and we propose avenues for improvements and cross-community engagement.

In the section below, we make a case for generative models as a successful and powerful technology which is inherently limited in its creative abilities by its uni-dimensional objective of perfection. The following section discusses how, in spite of its limitations, GANs have been used and abused as artwork production engines. We then explore how computational creativity research can contribute to further evolve such models into more autonomous creative systems, looking specifically at novelty measures as a first step towards this goal. We conclude by returning to the notion of bridging the two fields and describing future work.

Learning for Perfection

While the purpose of GANs, like all generative models, is to accurately capture the patterns in a data set and model its underlying distribution, guaranteeing convergence for this particular method remains a challenge (Lucic et al. 2018). Theoretical analyses of the GAN training objective suggest that the models fall significantly short of learning the target distribution and may not have good generalisation properties (Arora et al. 2017). It is further suggested that GANs in particular might be better suited for other purposes than distribution learning. Given their high-quality output and wide artistic acceptance, we argue for the adaptation of this generative approach for computational creativity purposes.

Generative models are currently only good at producing ‘more of the same’; their objective is to approximate the data distribution of a given data set as closely as possible. This highlights two sides of the same fundamental issue. First, in practice it remains unclear whether models with millions of parameters simply memorise and re-produce training examples. Performance monitoring through a hold-out test set is rarely applied and overfitting in generative models is not widely studied. Second, conceptually, such models are only of limited interest for creative applications if they produce artefacts that are insignificantly different from the examples used in training. Hence we further argue for an adaptation such that generative capabilities align with the objectives of computational creativity: to take on creative responsibilities, to formulate their own intentions, and to assess their output independently (Colton and Wiggins 2012).
Active Divergence

In order to produce artefacts in a creative setting, GANs still require expert knowledge and major interventions. Artists use a variety of techniques to explore, break and tweak, or otherwise intervene in the generative process. The following is a brief overview of some of those techniques. From a purely machine learning perspective, these exploits and accidents would be considered abuses and produce only sub-optimal results. Actively diverging from local likelihood maxima in a generator’s internal representation is necessary to find those regions that hold sub-optimal, but interesting and novel encodings.

Latent space search is a common practice among GAN artists, in which a neural network’s internal representation is explored for interesting artefacts. Traversing from one point to another produces morphing animations, so-called ‘latent space walks’. The space is often manually surveyed. Whenever precise evaluation criteria are available, evolutionary algorithms can be employed to automate the search for artefacts that satisfy a given set of constraints (Fernandes, Correia, and Machado 2020).

Cross-domain training forcefully mixes two (or more) training sets of the same type but different depictions, such that a model is first fit to the images from one domain (e.g. human faces) and then fine-tuned to another (e.g. beetles). The resulting output combines features of both into crossover images (fig. 1). Finding the right moment to stop fine-tuning is crucial and human supervision in this process is currently indispensable.

Loss hacking intervenes at the training stage of a model where the generator’s loss function is manipulated in a way that diverts it towards sub-optimal (w.r.t. the traditional GAN training objective) but interesting results. Given a model that generates human faces, for example, the loss function can be negated in a fine-tuning process such that it produces faces that the discriminator believes are fake (fig. 2; Broad, Leymarie, and Grierson 2020). Again, human supervision and curation of the results is just as important as devising the initial loss manipulation.

Early stopping and rollbacks are necessary whenever a model becomes too good at the task it is being optimised for. Akin to the pruning of decision trees as a regularisation method or focusing on sub-optimal (in terms of fitness functions) artefacts produced by evolutionary methods, rollbacks can improve generalisation, resulting in artefacts that are unexpected rather than perfect.

Summary

All of the above techniques require manual interventions that rely on human action and personal judgement. There are no well-defined general criteria for how much to intervene, at which point and by how much, or when to stop. It is central to an artistic practice to develop such standards, nurture their individuality and the difference to other practices. A major theme in GAN art, however, and a commonality in the above non-standard usages, is the active divergence from the original objective of the tool of the trade, in pursuit of novelty and surprise. This dynamic appears to be, in contrast to other artistic disciplines, exceptionally pronounced due to the use of state-of-the-art technology that has yet to find its definite place and purpose and whose capabilities are open to be explored. We celebrate and support this endeavour and argue that computational creativity can help by pushing generative models further, towards new objectives.

New Objectives for Generative Models

Two avenues of improvements for generative deep learning come to mind in a creative setting. First, we can consider creativity support tools, where a person is in charge of the creative process and the technology takes on an assistive role. For this, generative models need to be more accessible, as active divergence techniques still require highly...

---

1Tweet by @mmariansky
https://twitter.com/mmariansky/status/1226756838613491713
technical knowledge. They further need to be more controllable in their generative process. Active research on disentangled representation learning has recently proposed interpretable controls for global image manipulation (Härkönen et al. 2020). Common dimensions of variance in the data are first identified by the model and later manually sighted and named. Interpretable controls allow for the manipulation of images in a single specific aspect, such as a person’s age, the exposure of a photograph or the depicted time of day, while maintaining the others (fig. 3). Similarly, localised semantic image edits (Collins et al. 2020) transfer the appearance of a specific object part from a reference image to a target image, e.g. one person’s nose onto another person’s face.

Second, generative models are far from completely autonomous creative systems that are able to formulate an intention and assess their own output. As a start, these models require readjustments and extensions to be pushed from mere imitation to genuine creation. While creativity is arguably an essentially contested concept (Jordanous and Keller 2016) and there exist a variety of individual definitions, many of those include the notions of novelty, surprise and some form of value (e.g., usefulness or significance) (Jordanous 2013). In our analysis of GAN artists’ practices, a very clear commonality was the abusing of the standard practice in order to produce novel, perhaps surprising, outputs. Hence we will here focus on the aspect of novelty and on how the output of generative models could be assessed in regards to novelty.

Evaluating Novelty

As many evaluation schemes for creativity include notions of novelty, an exhaustive review of the literature is beyond the scope of this paper, as is the relationship and subtle differences between novelty and surprise (Macedo and Cardoso 2017). We focus here on explicit measures of novelty, in particular in the context of generative models. Currently, novelty can be achieved by tuning the stochasticity of a generative process whenever it is conditioned on a distribution of probabilities. In GANs, the latent code truncation trick clips values drawn from a normal distribution to fall within a limited range (Brock, Donahue, and Simonyan 2019). On the other end, a temperature parameter can be applied to scale a network’s softmax output (Feinman and Lake 2020). Both improve the quality of individual artefacts at the cost of sample diversity. While the original intention is to decrease randomness in order to obtain artefacts closer to the mean, they may also be able to achieve the inverse. Neural network-based methods have been proposed for the generation of novel artefacts, e.g., CAN (Elgammal et al. 2017), Combinets (Guzdial and Riedl 2019), as well as a number of metrics for the evaluation of GANs, e.g., the Inception Score (Salimans et al. 2016) and FID (Heusel et al. 2017). However, none of these can be used to measure novelty or to compare the extent to which deep learning methods are capable of producing it. For a measure that might fill this gap, we can draw from work in computational creativity.

Ritchie (2007) proposes a formal framework of empirical criteria for the evaluation of a computer program’s creativity, advocating for a post-hoc assessment based on a system’s output and independent of its process. A definition of creativity focuses on novelty, quality and typicality, where the latter refers to whether an artefact matches the intended class (e.g., when generating jokes, whether it has the formal structure of a joke). Quality (also denoted as value) and typicality are expressed as ratings, novelty is seen as the relationship between the input and output of a program and formalised in a collection of proportions in set-theoretic terms.

Most interesting for our purposes is Ritchie’s concept of an ‘inspiring set’, which could be treated as the knowledge base but, in the context of learning algorithms, does not have to be equivalent to the training set. Representing the examples that the author of a generative system hopes to achieve, it would be too trivial to allow a learning algorithm a glimpse at such examples. Rather, an inspiring set can inform about the necessary choices in the design process of a generative system that might evoke the desired output. Current discussions around the inductive biases of the fundamental building blocks in deep learning pose similar questions. Recent work has tried to leverage the specific choice of structure in hybrid neuro-symbolic models (Feinman and Lake 2020). This idea leaves room for the question of how the concept of an inspiring set could be integrated into the training and sampling schemes of a generative model.

In work on curious agents, Saunders et al. (2004; 2010) use Self Organising Maps (SOM) (Kohonen 1988), to measure the novelty of an input through a distance metric in vector space and in comparison to all other examples stored in the SOM. ‘Interestingness’ is estimated through an approximation of the Wundt curve (Berlyne 1960) (the sum of two sigmoids), to the effect that the score peaks at moderate values of novelty and rapidly drops thereafter. This model is based on the understanding that for new stimuli to be arousing, they have to be sufficiently different but not too dissimilar from known observations.

Pease, Winterstein, and Colton (2001) discuss novelty in relation to complexity, archetypes and surprise, and propose specific metrics for these aspects. First, an item is deemed more novel the more complex it is. Complexity is defined in terms of the size of a given domain and how unusual and
complicated the generation of an item is, which attempts to capture how many rules and how much knowledge was necessary in the process. Second, responding to Ritchie’s typicality, novelty is defined as the distance of an item to a domain’s given archetypes. This approach is similar to Saunders et al. (2004; 2010) in that it compares items to a knowledge base and computes distances in vector space. Third, the authors argue that ‘fundamental’ novelty evokes surprise as a reaction. However, a metric for surprise cannot be used to prove novelty, it only shows the absence of ‘fundamental’ novelty through the lack of surprise.

Conclusions and Future Work
The bridge between computational creativity (CC) and generative deep learning is currently one-way only. That is, CC researchers regularly draw on deep learning techniques in their projects, but the artificial neural network community rarely draws from the philosophy, evaluation or techniques developed in CC research, even for generative projects. The methodology for reversing the traffic presented here seems sound: survey ways in which artists use and abuse deep learning for creative purposes, identify how current practice limits this, and draw from CC research to address the shortcomings. For the bridge to be truly successful, any flow of information from CC into deep learning must respect the culture of the latter field. In particular, we will be aiming to develop concrete numerical evaluation methods for important aspects such as novelty, against which different models can be compared and progress shown, perhaps framing novel generation of artefacts as solving a problem of generating surprising results. This could lead to test suite data sets and potentially a kaggle.com competition, etc.

In the digital arts, deep generative models have found wide application as avant-garde tools, continuously demonstrating their potential. However, as these tools emerged from the discipline of machine learning, the objective of perfectly modelling patterns in data stands in the way of generative models further evolving towards autonomous creative systems. Active divergence is the common theme of a number of techniques we have explored, illustrating how GAN artists strive for sub-optimal solutions rather than perfect reproductions in the pursuit of novelty and surprise. These techniques, however, require much human intervention, supervision and highly technical knowledge which further limits their accessibility. We believe that computational creativity methods and methodologies, evaluation criteria and philosophical discourses can help progress deep generative learning so that non-standard creative usages become standard and the machine learning community embraces currently (seemingly) fuzzy ideas such as novelty and surprise. In the process, CC researchers will have access to increasingly powerful, autonomous and possibly creative techniques for exciting and ground-breaking projects.

Acknowledgements
We thank the anonymous reviewers. This work was supported by the EPSRC Centre for Doctoral Training in Intelligent Games & Games Intelligence (IGGI) [EP/S022325/1].

References
Arora, S.; Ge, R.; Liang, Y.; Ma, T.; and Zhang, Y. 2017. Generalization and Equilibrium in Generative Adversarial Nets (GANs). In Proceedings of ICML.
Brock, A.; Donahue, J.; and Simonyan, K. 2019. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In ICLR.
Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In Adv. NIPS.
Abstract

In this paper, we propose the demonstration of a system that changes country flags based on trendy topics retrieved from news titles. We give an overview of the system and introduce the reader to several topics on which the system has impact.

Introduction

Flags are among the symbols of a nation that help the formation and maintenance of a national identity, both internally—among its citizens—and externally—keeping a coherent sense of oneness in perception of other countries and entities. This process of maintaining a collective identity is described by Geisler (2005) as an “ongoing, dynamic process in which historical symbolic meanings are constantly recycled, actualised, challenged, renegotiated, and reconfirmed”.

In fact, it is possible to analyse the evolution of a flag and its transformations, relating them to changes in the entity that the flag stands for (e.g. political changes in the country). Moreover, by looking at country flags one can easily identify similarities among them, which point to how different flags influenced each other throughout history (Healy, 1994). The exploration of this relational character is observed in imaginary scenarios, for example an alternate universe in which Nazi Germany and Imperial Japan won the World War II. This scenario is depicted in the Amazon’s mini-series “The Man In The High Castle” (Heller, 2015), resulting in the design of fictional flags for an America ruled by Nazi and Japanese forces.

Going beyond the reflection of its evolutionary path, a flag exerts its most significant role as a means of conveying the intended image of the entity that it stands for. One example is design of a new European Union by Rem Koolhaas based on the essence of the European project as a joint effort of different nation states, each with its own identity but together contributing to a plural identity of EU. The redesign resulted in a barcode-style flag featuring the colors of EU countries, transmitting the idea of individual identities and simultaneous advantages of acting together.

On the other hand, the sense of identity has also fragilities. The value of one’s identity makes it so that it is often prone to exploitation and manipulation, for example by the misappropriation of flags. The Double Standards project (Pater, 2012) investigated 59 seajacked ships that mask their owner’s nationality by purchasing a “cheap flag” from another country to avoid taxes and environmental regulations. Such examples highlight how volatile an identity can be, especially in a time when individual identity loses power to the growing advances of globalisation. Moreover, in addition to this dissolution of individuality, in the current society characterised by constant change, the idea of an immutable identity becomes more and more questionable.

This sense of fluid identity is explored, for example, in the

---

Figure 1: Flags generated on November 15th 2019 and July 15th 2020. Below each flag, the country of the original flag and the topic used in the generation are identified.
project Net.flag (Napier, 2002), which is based on the idea of an “ever-changing flag of the Internet” that anyone can alter upon visiting its website. In our opinion, these issues are ground for an important discussion on how the identity of a nation is represent by its flag and on the impact brought by changes in this national symbol. The system that we propose to demonstrate, initially presented in (Cunha et al., 2020), aims at confronting the viewer with questions of identity, by altering country flags based on current events (see Fig. 1).

This paper has two main goals: (i) propose a system demonstration at the Eleventh International Conference on Computational Creativity (ICCC’20) and (ii) further develop on the topic of impact and ethical considerations of our system.

**Mutable flags**

The rate at which society changes as well as the access to global information have been increasing. One can question whether a nation only possesses an identity or, based on this constant change, if it can also be assigned what we refer to as “mood” – i.e. what is happening in the country at the moment. This concept is aligned with a flag campaign that was made for “Grande Reportagem Magazine” in 2005, in which the meanings of seven flags were changed based on shocking facts about the country. As such, the following question is posed: Can the mood of a country be represented in its flag?

Most projects that address flag generation either consist of flag editors based on pre-defined grammars – e.g. the web app Scratchon’s Flag Designer by Lars Ruoff – or flag generating software such as Twitter bots – e.g. the Flags Mashup Bot produces flags that are the visual blend between two existing flags.

Three main aspects can be identified as having particular importance in an individual flag: (i) structure (i.e. how it is divided, what elements it includes, etc.); (ii) meaning associated with its elements; and (iii) what the flag symbolises (e.g. a national flag represents a nation). Nonetheless, most approaches are centred on aspects related to the structure of the flag and disregard meaning. Our system links both aspects and uses them to affect the third – what the flag represents.

Our goal is to represent what we refer to as “mood” of the country, using the flag of a country as the starting point and applying changes according to real-time data about the country. The concept of “mood” is based on the expression I’m in the mood for [something], which is normally used in association with feelings that do not last long. This reactivity to external input can instil a quality of “being alive” into the flag (Martins et al., 2019), which is in full alignment with our goals.

**System Overview**

In this section, we provide a general overview of the system. As the system was thoroughly described in (Cunha et al., 2020), we will refrain from going into much detail.

**Generating Flags** The system relies on two base assumptions: (i) when generating of a flag, an existing flag would be given as input and (ii) the changes made should allow the initial flag to be recognised. The produced flag should be perceived as a transformation to the original one, thus allowing the observer to identify the country.

The first step of the flag production process consists in searching elements that match a queried word. These are then used to change the initial flag. The element search is conducted in three places:

- **Existing Flags**: we produced a flag dataset that included both visual data and semantic data (see Fig. 2). Using this dataset, a search for the input word is conducted on the meanings associated to elements of existing flags;

- **Colour Names**: we merged existing datasets to produce a list of 3,476 colours and associated names (e.g. #ef4026 has the name “Tomato”). This list is used to search for the input word;

- **Emoji**: we use the EmojiNet (Wijeratne et al., 2017) dataset to find emoji based on the input word, similar to what is done in the Emojinating system (Cunha et al., 2019).

The transformation made to the flag depends on the type of elements found. For example, if the input word is found on a colour name, the colour is applied to an element of the base flag. On the other hand, if the element found is an emoji, it can either be added to the flag or replace an existing symbol.

**Trend-driven Flags** The notion of “mutable flag” gains even more significance when combined with a sense of reactivity – something is “reactive” when it changes according

---

Figure 2: Data collected for the Cyprus flag: id assigned (e.g. cy-island), description (“copper island”) and meanings (M stands for general meaning, MC for meaning of colour and MS for meaning of shape).

2http://creativecriminals.com/print/grande-reportagem/flags
3http://flag-designer.appspot.com/
4https://twitter.com/FlagsMashupBot/
to external input. Our main goal was to generate flags that changed according to current events, instilling a quality of “being alive” into them. As such, when generating a flag for a given country, the latest news titles in English that mention the name of the country are automatically retrieved by the system, which then uses them to identify the most predominant topics (nouns). To do this, we gather news titles from Google News RSS feed and conduct a Part-of-Speech tagging with the Javascript tagger jspos\(^3\), to extract nouns from the initial news titles and identifying the most predominant ones. Based on the most predominant ones, we then generate flags that represent the “mood” of the country.

**Producing Explanations** Flags have a huge layer of symbolism as most of their elements have associated meanings. When generating a flag, such layer is also of great importance. As such, in addition to producing flags, we also produce explanations that provide clues of how and why the flag was changed (see Fig. 3). This establishes a link between the visual output and the reasoning based on which the flag was produced, thus making evident the conceptual foundation of the generation process. Explanations follow a predefined structure:

\[
\text{[element X] represents/stands for/symbolises [Y]}
\]

where Y is the queried word and X depends on the change nature.

The act of providing explanations is aligned with the guidelines for explainability\(^5\) in AI Ethics (Jobin, Ienca, and Vayena, 2019), which has been seen as having an important role in design systems – e.g. Zhu et al. (2018) focus on explainability and provide guidelines on how it can be applied in game design. Moreover, the production of explanations can be interpreted as a process of Framing as defined by Cook et al. (2019):

“Framing’ refers to anything (co-)created by software with the purpose of altering an audience or collaborator’s perception of a creative work or its creator.

This process plays an important role in how unbiased observers perceive AI systems and their output. Furthermore, Cook et al. (2019) state that implementing methods for the systems to explain themselves can improve the relationship between user and AI agent. Framing is described as having three aspects: sources of information (e.g. where the meanings of the flags are retrieved from and on what are they based), means of framing (e.g. providing descriptions in natural language for the produced flags) and purposes for framing (i.e. the intended impact on the audience). Regarding this last aspect, the main motivation behind the development of our system is to have an impact on the observer, as opposed to a generation for mere aesthetic purposes. In the following section we provide more detail on this subject.

**Ethical Considerations And Potential Impact**

Our goals go beyond the mere generation of flags. In fact, our main motivation was that the system and the results that it produces would have an impact on the audience.

The limits of use of a national flag have long been a topic of debate. As we have seen, flags are prone to be misappropriated – aspect highlighted in The Double Standards project (Pater, 2012). Moreover, as symbols of a nation they are often used in acts motivated by political reasons – flags being burnt in protests. For these reasons, several cases exist of controversy around what is considered legal and what is to be seen as flag desecration (Goldstein, 2019; Marinthe et al., 2019). However, the limits are often blurry and lead to strong yet opposing reactions when they are tested. One example is the installation *What is the Proper Way to Display a U.S. Flag?* by Dread Scott\(^7\) which showed two images featuring the American Flag, one of which displayed a flag being burnt, and encouraged the audience to write responses to the question in the installation’s title. Upon writing a response, the audience had the option of standing on the flag. The installation triggered very strong reactions – from thank you messages to death threats. But more importantly, led to a discussion on what is a misuse of the flag and the legality of such. Obviously, there is a great distance difference by purposely destroying a flag and using it to pass an idea. The latter being especially important for artistic purposes (Hartvigsen, 2018). Focusing on what we are proposing in this paper, to what extent do flags actually represent constantly evolving nations when they are subject to rules often against change and transformation? In addition, people are not always receptive to changes in the national flag as it deals with questions of their own identity (Osborne et al., 2016). This immutability reaches the point that the flag design stays the same but the meanings change – e.g. the colours of the Portuguese flag went from a political connotation (party colours) to more general ones (e.g. green being associated “hope”).

\(^3\)https://code.google.com/archive/p/jspos/  
\(^5\)Our interpretation of the principle of explainability and transparency is based on the description from AI for People https://www.aiforpeople.org/  
\(^7\)https://www.dreadscott.net/
We intend to contribute to this discussion by questioning the unchangeable status of a flag. As such, we identify several topics that we believe our system has the potential to have impact on:

- **Own sense of identity**: the feelings towards a flag vary from person to person: some might not have a big connection to this symbol; others might proudly display it on the window to convey a sense of national support (e.g. in some countries flags are often hanged from windows in support of the national soccer team); and, possibly, there may be citizens that feel misrepresented by the flag (Wright, 2011). In any case, we believe that changing a country’s flag will lead to a sense of “discomfort” by creating a gap between the original symbol and an altered version, possibly making people wonder if they still identify themselves with it;

- **Evolution of daily topics**: flags are often objects that have a very slow evolution – they stay the same for long periods. Our system brings changes in this regard by allowing flags to adapt to current events. Such approach enables the user to observe a constant change in the flag, a consequence of changes in the “mood” of the country;

- **Event Highlighting**: despite living on what can be called a “global village”, there are many events that often go unnoticed, even though they deserve our utmost attention – an example is given in (Cunha et al., 2020) regarding a huge oil spill that was not widely known. Our system as the potential of being exploited as a visualisation tool with the goal of highlighting such events. Using flags to call the public attention has been explored in the past, e.g. (Pater, 2012).

**Conclusion**

The goal of this paper is to propose the demonstration of the system initially presented in (Cunha et al., 2020). As such, we only provided a general overview on the system and identified topics that are address by it. A video of the system being used can be seen at https://rebrand.ly/iccc20demo.

**Acknowledgements** This work is partially funded by the FCT - Foundation for Science and Technology, I.P., within the scope of the project C93CU - UID/CEC/00326/2020 and by European Social Fund, through the Regional Operational Program Centro 2020, and under the grant SFRH/BD/120905/2016.

**References**


7. Visual Creativity
Abstract

From classifying handwritten digits to generating strings of text, the datasets which have received long-time focus from the machine learning community vary greatly in their subject matter. This has motivated a renewed interest in building datasets which are socially and culturally relevant, so that algorithmic research may have a more direct and immediate impact on society. One such area is in history and the humanities, where better and relevant machine learning models can accelerate research across various fields. To this end, newly released benchmarks (Clanuwat et al. 2018) and models (Clanuwat, Lamb, and Kitamoto 2019) have been proposed for transcribing historical Japanese cursive writing, yet for the field as a whole using machine learning for historical Japanese artworks still remains largely uncharted. To bridge this gap, in this work we propose a new dataset KaoKore which consists of faces extracted from pre-modern Japanese artwork. We demonstrate its value as both a dataset for image classification as well as a creative and artistic dataset, which we explore using generative models.

Introduction

In pre-modern Japan, one well-established narrative form consisted of stories displayed entirely as one long continuous painting, usually in the form of a picture scroll (絵巻物, Emakimono) or a picture book (絵本, Ehon), accompanying cursive writing of the story itself. These stories include diverse arrays of characters (see Figure 1), and thus provide valuable materials for the study of Japanese art history.

In art history research, comparative style study, based on the visual comparison of characteristics in artworks, is a typical approach to answering research questions about works, such as the identification of creators, the period of production, and the skill of the painter. Among many attributes that appear in artworks, facial expressions offer especially rich information not only about the content but also about how the artworks were created. To accelerate comparative studies, Collection of Facial Expressions (Suzuki, Takagishi, and Kitamoto 2018) has been created as a collection of cropped faces from multiple artworks with basic metadata annotated manually by art history experts. It was made possible leveraging recent technological developments, such as mass digitization of historical Japanese artworks and image sharing using IIIF (Kitamoto, Homma, and Saier 2018), implemented using JSON-LD, a JSON serialization pattern.

Inspired by the recent success of developing new benchmarks (Clanuwat et al. 2018) as well as new models (Clanuwat, Lamb, and Kitamoto 2019) in the field of Japanese cursive writing, we believe that Collection of Facial Expressions provides a largely unexplored opportunity to

---

Figure 1: A cropped page of Tale of the Hollow Tree (宇津保物語, Utsuho Monogatari), a late-10th century Japanese story, presented in the form of a picture book (絵本, Ehon) in the 17th century (Unknown Late 10th Century). Picture scrolls and picture books usually have cursive texts telling the story (left) in addition to story-explaining paintings depicting many characters (right), often on the same page.
bridge research of historical Japanese artwork with machine learning research. It, however, is not designed for machine learning in terms of data format (JSON-LD), image size (of different and irregular sizes) and attribute values (of both related and unrelated sets). As a result, it presents an obstacle for easy adaption of machine learning techniques.

To bridge this gap, in this work we propose a novel dataset, KaoKore, derived from the Collection of Facial Expressions. Our contributions can be summarized as follows:

- We process the Collection of Facial Expressions to create the new KaoKore dataset of face images from Japanese artworks along with multiple labels for each face in a more simple, regular and easy-to-use format.
- We provide standard data loaders for both PyTorch and TensorFlow as well as official train/dev/test splits which make this dataset easy to use across frameworks and compare results with.
- We demonstrate the dataset’s utility for image classification by introducing a number of new baselines.
- We study how different types of generative models can be applied to this dataset and support different types of creative exploration.

We hope that this dataset will help to strengthen the link between the machine learning community and research in Japanese art history.

**KaoKore Dataset**

We begin with describing Collection of Facial Expressions with is the foundation on which we build our work, the KaoKore dataset. Collection of Facial Expressions results from an effort by the ROIS-DS Center for Open Data in the Humanities (CODH) that has been publicly available since 2018. Collection of Facial Expressions provides a dataset of cropped face images extracted from Japanese artwork from the Late Muromachi Period (16th century) to the Early Edo Period (17th century) to facilitate research into art history, especially the study of artistic style. It also provides corresponding metadata annotated by researchers with domain expertise. An example of our cropping process is shown on the left panel of Figure 2.

The Collection of Facial Expressions is built upon the International Image Interoperability Framework (IIIF) and IIIF Curation Platform (Kitamoto, Homma, and Saier 2018),

Figure 3: Exemplary images in the KaoKore dataset. These examples demonstrate various faces depicting different subjects in diverse yet coherent artistic styles.

---

2Publicly available from National Institute of Japanese Literature, Kyoto University Rare Materials Digital Archive and Keio University Media Center.
which is a system designed to ease the burden of common tasks in humanities research such as viewing, searching, and annotating documents, by representing documents in a structured and machine-readable data format. Collection of Facial Expressions serves as the basis for several research projects in art history. One example, shown on the right panel of Figure 2, is the comparison of similar-looking faces from different artworks. This type of art history research provides insights into the history of trending art styles and is useful for determining authorship of works.

We derive KaoKore dataset (see Figure 3 for exemplary images) from Collection of Facial Expressions in a form that will be easily recognizable to the machine learning community, with the hope of facilitating dialogue and collaboration with humanities researchers. Concretely, we process the images and labels into industry-standard formats such that the resulting dataset is easy to use with off-the-shelf machine learning models and tools. Since the cropped faces from the Collection of Facial Expressions can have different sizes and aspect ratios, we pre-process them to ensure that all cropped images are normalised to the same size and aspect ratio.

The resulting KaoKore dataset contains 5552 RGB image files of size 256 x 256. Figure 3 shows examples from the KaoKore dataset, a collection of various faces in diverse yet coherent artistic styles. The format follows that of ImageNet (Deng et al. 2009), making it a potential drop-in replacement dataset for existing machine learning setups.

To facilitate supervised learning, we provide two sets of labels for all faces: gender and (social) status, both from the frequently appearing subset of all expert-annotated labels from the Collection of Facial Expressions. Table 1 shows the classes and labels available in the KaoKore dataset, with exemplary images for each label. We setup tasks on the labels that appear most frequently. For example for class (social) status we choose noble, warrior, incarnation and commoner which each has at least 600 images, while discarding rare ones like priesthood and animal, each having merely a dozen. This is to avoid an overly unbalanced distribution over the labels. We also give official training, validation, and test sets splits to enable model comparisons in future studies.

<table>
<thead>
<tr>
<th>Class</th>
<th>Labels</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>gender</td>
<td>male</td>
<td><img src="image1" alt="Male example" /> <img src="image2" alt="Male example" /> <img src="image3" alt="Male example" /></td>
</tr>
<tr>
<td></td>
<td>female</td>
<td><img src="image4" alt="Female example" /> <img src="image5" alt="Female example" /> <img src="image6" alt="Female example" /></td>
</tr>
<tr>
<td>status</td>
<td>noble</td>
<td><img src="image7" alt="Noble example" /> <img src="image8" alt="Noble example" /> <img src="image9" alt="Noble example" /></td>
</tr>
<tr>
<td></td>
<td>warrior</td>
<td><img src="image10" alt="Warrior example" /> <img src="image11" alt="Warrior example" /> <img src="image12" alt="Warrior example" /></td>
</tr>
<tr>
<td></td>
<td>incarnation</td>
<td><img src="image13" alt="Incarnation example" /> <img src="image14" alt="Incarnation example" /> <img src="image15" alt="Incarnation example" /></td>
</tr>
<tr>
<td></td>
<td>commoner</td>
<td><img src="image16" alt="Commoner example" /> <img src="image17" alt="Commoner example" /> <img src="image18" alt="Commoner example" /></td>
</tr>
</tbody>
</table>

Table 1: Labels available in the dataset along with exemplary images belonging to each label.

<table>
<thead>
<tr>
<th>Model</th>
<th>classifying gender</th>
<th>classifying status</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG-11</td>
<td>92.03 %</td>
<td>78.74 %</td>
</tr>
<tr>
<td>AlexNet</td>
<td>91.27 %</td>
<td>78.93 %</td>
</tr>
<tr>
<td>ResNet-18</td>
<td>92.98 %</td>
<td>82.16 %</td>
</tr>
<tr>
<td>ResNet-34</td>
<td>93.55 %</td>
<td>84.82 %</td>
</tr>
<tr>
<td>MobileNet-v2</td>
<td>95.06 %</td>
<td>82.35 %</td>
</tr>
<tr>
<td>DenseNet-121</td>
<td>94.31 %</td>
<td>79.70 %</td>
</tr>
<tr>
<td>Inception-v3</td>
<td>96.58 %</td>
<td>84.25 %</td>
</tr>
</tbody>
</table>

Table 2: Test accuracy on classification tasks. See text for the discussion and citation for each model.
Experiments

We conduct two types of experiments. First, we provide quantitative results on the supervised machine learning tasks of gender and status classification of KaoKore images. Second, we provide qualitative results from generative models on the KaoKore images.

Classification Results for KaoKore Dataset

We present classification results for the KaoKore dataset in Table 2 using several neural network architectures, namely VGG (Simonyan and Zisserman 2014), AlexNet (Krizhevsky 2014), ResNet (He et al. 2016), MobileNet-V2 (Sandler et al. 2018), DenseNet (Huang et al. 2017) and Inception (Szegedy et al. 2016). We use PyTorch’s reference implementation (Paszke et al. 2019) of common visual models, standard data augmentation, and Adam optimizer (Kingma and Ba 2014). We use early stopping on the validation set, and report the test set accuracy.

In providing accuracies across various models we demonstrate that standard classifiers are able to achieve decent but imperfect performance on these tasks. Additionally, we show that newer and larger architectures often achieve better performance, which suggests that even further improvement through better architectures will be possible.

Creativity Applications

As the KaoKore dataset is based on artwork, we also investigate its creative applications. While our hope is that people will find novel ways of engaging with this dataset artistically, we demonstrate that reasonable results can be achieved using today’s best-performing generative models. We note that faces in the KaoKore dataset contain many correlated
attributes like face shape and hairstyle, giving a challenging tasks of correctly model correlations, yet are highly stylized and simpler than realistic images of faces, creating challenges in modeling data distribution. Since these two challenges are non-trivial for generative models, we hope the KaoKore dataset will be useful for generative modeling research.

Generative Adversarial Networks We first explore Generative Adversarial Networks (GANs) (Goodfellow et al. 2014) which have been successfully used as generative models for synthesizing high quality images (Karras et al. 2017; Karras, Laine, and Aila 2019; Zhang et al. 2018), and has also seen creative applications such as image-to-image translation (Zhu et al. 2017), controllable Anime character generation (Jin et al. 2017; Hamada et al. 2018; Jin and others 2020), photo-realistic face generation (Lu, Tai, and Tang 2018), apartment (Chaillou 2019), and fashion design (Chen et al. 2020).

Inspired by this, we leverage StyleGAN (Karras, Laine, and Aila 2019), a state-of-the-art GAN model for images. We implement and train it on our dataset and show the resulting generated images. In Figure 4, we show uncurated images produced by StyleGAN, showing that the varieties in our datasets are successfully captured by the generative model.

Neural Painting Models We find that GAN models are able to generate somewhat plausible-looking images of KaoKore faces (Figure 4). However, the GAN objective requires that the model directly generate pixels, where an artist would paint the image by applying strokes it-
eratively on a canvas. Thus when a GAN makes mistakes, the types of errors it makes are generally quite unlike the variations that could be produced by a human painter. To give the synthesis process a more artwork-like inductive bias, we consider Stroke-based rendering (Hertzmann 2003) which produces a reference painting by sequentially drawing primitives, such as simple strokes, onto a canvas. Recent advances using neural networks have been proposed by integrating techniques such as differentiable image parameterizations (Mordvintsev et al. 2018; Nakano 2019) or reinforcement learning(Ganin et al. 2018; Huang, Heng, and Zhou 2019), which can greatly improve the quality of generated painting sequences. Given that our proposed KaoKore dataset is based on art collections, we are interested in applying these neural painting methods with image production mechanisms that better resemble human artists. In particular, we explore applying intrinsic style transfer (Nakano 2019) and learning to paint (Huang, Heng, and Zhou 2019) on the proposed KaoKore dataset.

**Intrinsic style transfer (Nakano 2019)** is a set of methods combining differentiable approximation with non-differentiable painting primitives (e.g. strokes in the colored pencil drawing style) and an adversarially trained neural agent that learns to recreate the reference image on the canvas by producing these primitives sequentially. It is characterized by a lack of “style-loss” that are often used in style transfer methods to carry the reference image’s style into the target one, which in turn exposes the intrinsic style derived from painting primitives mentioned above. In Figure 5, we show the produced painting sequences on a few exemplary images. It can be observed that the image has been decomposed into strokes that resemble how a human artist might create the pencil sketch, while the model has not been provided with any recording of sketching sequence. This is further epitomized in Figure 6 and Figure 7, which show the reference images and the final canvas after completing painting.

**Learning to paint (Huang, Heng, and Zhou 2019)** is a neural painting model which differentiates itself from others in a few aspects, including using regions marked by quadratic Bézier curves as painting primitives as well as leveraging model-based deep reinforcement learning for training. As a result, its painting sequence is made of simple regions rather than brush-like strokes, and the sequence is as short as possible due to the optimization goal used in reinforcement learning training. As shown in Figure 8, 9 and 10, the method learns to assemble simple curve regions in recreating the image that emphasize the general arrangements of objects in the painting and resembles an abstract painting style.

The two neural painting models explored can, given a single image from the KaoKore dataset, produce painting sequences that could be applied on a (virtual) canvas and resemble human-interpretable styles. Yet due to each method’s fundamentally different mechanism, the style, while being expressive, as a result also resort to different artistic style. By simultaneously presenting artistic familiarity in the style and surprising in how to decompose a single image, we hope this result can provide insights into the study of art styles.

**Discussion and Future Work**

We hope that our proposed KaoKore dataset provides a foundation on which future works can be built: including humanities research of historical Japanese artworks or machine learning research with creative models, given our dataset’s dual value both as a classification dataset as well as a creative and artistic one. In the future, we plan to increase the number of facial expression images in our dataset by building a machine learning powered human-in-the-loop annotation...
mechanism that allows us to scale the labeling process. We would also like to construct new datasets in future work which help to expand machine learning research and its applications for more general Japanese art. Finally, we anticipate that further interdisciplinary collaboration between humanities research and the machine learning research community would contribute to better and more efficient cultural preservation.

References


Unknown. Late 10th Century. Tale of the hollow tree (宇津保物語, utsuho monogatari). National Institute of


Uncovering Aesthetic Preferences of Neural Style Transfer-Generated Images with the Two-Alternative-Forced-Choice Task

Chaehan So
Information & Interaction Design
Humanities, Arts & Social Sciences Division, Yonsei University
Seoul, South Korea
Email: cso@yonsei.ac.kr

Abstract

Neural style transfer is a popular deep learning algorithm to generate images to mimic human artistry. This work applies the psychological method of the two-alternative forced choice (2afc) task to measure aesthetic preferences for neural style generated images. Portrait photos of three popular celebrities were generated by varying three parameters of neural style transfer in five configuration levels. Participants had to choose the image they preferred aesthetically from all pairwise combinations of configurations per style. The rate of being chosen was calculated for each neural style transfer configuration level. The findings show a differentiated picture of aesthetic preferences. On the one side, they indicate that people prefer images rendered with 500 iterations and a learning rate of 2e1, i.e. configurations that allow them to recognize the structure of the portrait image despite the stylization. On the other side, aesthetic preferences peak for two distinctly different content-to-style weight ratios. Whereas the medium-high configuration (100:100) may be favored by people who like abstract arts, the high configuration (300:100) may be chosen by people who prefer realistic art. These results indicate that aesthetic preferences for neural style transfer-generated images can be characterized by unique patterns, and their optimal configuration levels can be captured by the 2afc task.

Keywords: Neural style transfer, aesthetic preferences, 2afc task, parameter configuration

Introduction

Neural style transfer has become a popular generative deep learning algorithm for generating creative images by merging the structure of a visual input with the style of another image. Although the output is often called “creative”, the literature lacks a differentiated view on how visually pleasing humans perceive this stylized image. Historically, the development of the neural style transfer algorithm built on substantial progress of convolutional neural networks (CNNs) and generative adversarial networks (GANs). Gatys, Ecker, and Bethge (2015) introduced the former algorithm to show how the artistic style of painters can be transferred to another image. The neural style algorithm extracts the structural information of the input image, learns the color and texture information in the style image, and then renders the semantic structure of the input image in the color and texture of the style image (Gatys, Ecker, and Bethge 2016). Originally, neural style transfer was demonstrated with common photo motifs like houses or landscapes (Gatys, Ecker, and Bethge 2016). Neural style transfer has later been applied to doodles (Champandard 2016), videos (Huang et al. 2017), artistic improvisation (Choi 2018), and fashion (Jiang and Fu 2017; Zhu et al. 2017). Recent developments of neural style transfer include using pretrained models for stylization in so-called feedforward networks (Chen and Schmidt 2016). One such approach (Johnson, Alahi, and Fei-Fei 2016) leverages the use of the pretrained models to calculate losses on high-level features (so-called perceptual losses) instead of per-pixel losses. The generation of neural style transfer is configurable by many parameters. Therefore, it may be insightful which of these parameters configurations lead to aesthetically pleasing results.

Aesthetic Preferences

What image looks good? Psychological research has yielded manifold insight into this question from the investigation of interindividual differences in aesthetic taste. For example, aesthetic preferences are not stable across the human lifespan but follow an inverted U-shape peaking around early to middle adulthood (Pugach, Leder, and Graham 2017). The preference for different art genres depends on personality traits, e.g. people with high scores on trait openness tend to prefer abstract arts over renaissance art (Pelowski et al. 2017). People prefer a size of a displayed object which is larger (smaller) relative to the frame for larger (smaller) objects, and the preferred displayed object size is proportional to the logarithm of its physical size (Linsen et al. 2011).
Two-alternative forced choice

This study investigates people’s perception of artistry in an image with the two-alternative-forced choice (2afc) task. The 2afc task is an experimental method in psychology first introduced by Thurstone (1927). The 2afc task is frequently used in cognitive psychology to detect perception thresholds or evaluate the psychological differentiation of stimulus variation. Applying signal detection theory, the Thurstonian model of item-response theory identifies the detection thresholds of perceived stimuli in a 2afc task by fitting a maximum-likelihood estimator, the psychometric function, to the averaged 2afc task responses. In such 2afc applications, the presented image pairs contain a baseline image, i.e. an image without any effect, to which the other image is compared.

The 2afc task is a paired comparison test, i.e. it specifies the assessment of two samples. This simple design has shown the advantage of significantly reducing fatigue, carryover and memory effects encountered when assessing more samples (Yang and Ng 2017), and reducing the required sample size. Importantly, both samples in a 2afc task must be presented at the same time (Macmillan & Creelman, 2004, p. 148). If this condition is not met (one stimulus is shown), it represents a yes/no task (often used for lexical decisions like word/non-word) not to be confused with a 2afc task.

The 2afc task has also been used in psychological research to investigate aesthetic preferences, e.g. the classification of images as art vs. non-art (Pelowski et al. 2017), the spatial composition in multi-object pictures (Leyssen et al. 2012) (Leyssen et al. 2012), or the size of images for real-world objects (Linsen et al. 2011).

According to Palmer, Schloss, and Sammartino (2013), the common procedure for the 2afc for testing aesthetic preferences is to present the participant with all possible pairs of stimuli instead of all comparison with a baseline image. For each pairwise combination, participants are asked which they “like better” which corresponds to their aesthetic preference. The global measure of an image’s relative preference is calculated by the average probability (or actual count) of selecting it over all other images.

Research Question

Based on the increasing proliferation of neural style transfer applications, it becomes interesting from a design and art perspective how it can produce aesthetically pleasing results. Understanding related mechanisms leads to the investigation of available parameter configurations of neural style transfer. Such an investigation can unveil important underlying factors and relationships of human aesthetics perception. It may reveal for example that for certain parameters of neural style transfer, the preference curve may have a curvilinear (inverted-U or V) shape rather than a monotonous positive or negative gradient.

The preceding considerations lead to the following research question.

RQ: Which parameter configurations of neural style transfer reflect the highest aesthetic preference?

Method

Participants

This study recruited a sample of 18 participants undergraduate and graduate students of interaction design, as well as professional UX designers.

Participants were 50% male and 50% female, and on average 30.5 (SD = 8.52) years old.

The academic status was 33.3 % Bachelor student, followed by 27.8% Master student, 5.56 % professional with Bachelor’s degree, 27.8% professional with Master’s degree, and 5.56% professional with Ph.D. degree.

Participants’ nationality was mostly South Korean (77.8%) with one person each from Canada, Germany, Singapore, and the United States.

Neural Style Transfer Algorithm

The neural style algorithm merges the structural information of an input image with the color and texture of a style image. This ensures that the input image’s structural information (like face and body line structure) can be recognized in the output image.

The present work uses the implementation of neural style transfer provided by the Github repository jcjohnson/fast-neural-style (Johnson 2016). It is implemented in torch (Collobert et al. 2018) and provides an improved version of the neural style transfer algorithm of the Github repository jcjohnson/neural-style (Johnson 2015). The latter version implements the original optimization-based algorithm introduced by Gatys, Ecker and Bethge (2015).

The optimization-based algorithm also provided in the jcjohnson/fast-neural-style Github repository by the script “slow_neural_style.lua” and allows many configuration options. One can configure the options num_iterations (number of iterations processed), save_every (image generation after save_every iterations), GPU (use GPU or CPU).

To modify the stylization outcome, one can configure many detailed options of the loss network. The interface allows determining the content layers, style layers, style target (gram matrices vs. spatial average) and the choice of the optimizer (commonly LBG-S or Adam algorithm). All these options require a lot of expertise to understand the direction and magnitude of parameter changes. Configuration parameters with directly perceivable output impact include the number of iterations, the learning rate and the ratio between content weights and style weights.

All results of this paper were generated on an Ubuntu 16.04 LTS virtual machine in the Google Cloud. Image processing becomes impractical in CPU mode due to slow processing speed. Therefore, the present work used an Nvidia Tesla P100 GPU with CUDA 9.1 and CUDNN 8.0 libraries that offered a processing speed of a factor of approximately 100 times faster than a contemporary laptop with 8 CPUs.

Input Images. The stimuli were based on three faces of popular celebrities (Charlie Puth, Jessica Alba, Ellie Goulding). Figure 1 shows these input images.
Preliminary experiments provided the insight that pictures with high variance in the background were evaluated as part of the foreground by the algorithm and thus stylized the same way as the foreground. To avoid this effect, portraits were only selected if they showed a clear separation to the background.

**Style Images.** The style images were chosen from three different categories: black & white patterns, cloth design, and abstract arts. The preliminary experiments revealed that style images have the most impact on the output image if they contain texture information of finer granularity that separates well from the background. Figure 2 shows the set of style images used in this study.

**Parameters**

The neural style transfer was applied to the set of input images to create variations that were subsequently tested by 2afc tasks with the following parameters and configuration levels.

**Number of iterations.** The neural transfer algorithm by Gatys, Ecker and Bethge (2015) applies the L-BFGS optimizer on forward and backward iterations through the VGG-16 loss network. Johnson, Alahi and Fei-Fei (2016) found that the optimization is successful within 500 iterations in most cases. Therefore, this study compared generated visual results with 100, 200, 300, 400, and 500 iterations.

**Learning Rate.** The framework allows changing the optimizer from LBFGS to Adam. The Adam optimizer can be configured by the learning rate. This parameter specifies the step size in which weights are updated during the optimization. As Ruder (2016) points out, a too small learning rate slows down the convergence to the minimum, whereas an overly high learning rate can cause fluctuation around the minimum and thus hinder convergence.

The default learning rate is set to 1e-3. Therefore, this study explored the impact of learning rates for the Adam optimizer for 0.5e1, 1e1, 2e1, 4e1, and 6e1.

**Content-to-Style-Weight Ratio.** The content-to-style-weight ratio determines the degree of importance for the input image vs. the style image for rendering the output image. The default ratio is 1:5, i.e. the style weights are five times larger than the content weights. The parameter content weight is set as a relative value to style weight. The content-to-style-weight ratio defines the degree of importance given to the input image vs. style image for rendering the output image.

The default setting is 1:1 or 100:100. Therefore, this study explored results for the content-to-style-weight ratios 10:100, 50:100, 100:100, 200:100, and 300:100.

**2afc Task**

**Stimuli.** The preceding specifications generated the following three sets of stimuli.

1. Figure 4 shows Charlie Puth in variations of the number of iterations with values 100, 200, 300, 400, 500.
2. Figure 5 shows Jessica Alba in variations of the learning rate with values 0.5e1, 1e1, 2e1, 4e1, and 6e1.
3. Figure 6 shows Ellie Goulding in variations of the content-to-style-weight ratio with values 10:100, 50:100, 100:100, 200:100, and 300:100.

**Condition Counterbalancing.** The position of the two images in the 2afc task (either left or right) was counterbalanced within participants so that each image was evaluated in the same frequency in the left and right position.

**Pairwise Comparison Reduction.** The number of pairwise comparisons was reduced to reduce survey fatigue by using each image pair comparison only once, and balancing the image position only in absolute numbers rather than for each image pair combination.

**Procedure.** Before starting the survey, participants were asked to not use a smartphone but a laptop, desktop or tablet for larger image display. They were instructed to make the choice between the two images intuitively rather than by objective criteria. Each participant assessed 5 configurations of 3 parameters for 5 styles. Each configuration was assessed in 4 pairwise comparisons per style and participant, hence 72 times by all participants. Participants were encouraged to take a break every 50 trials for 10-20 seconds for stretching their upper body and relax their eyes.
Figure 4: Neural style transfer-variations of parameter A: Number of iterations

Figure 5: Neural style transfer-variations of parameter B: Learning rate

Figure 3: Neural style transfer-variations of parameter C: Content-to-style-weight ratio
Results

The findings are displayed in Figure 7 and show an aggregated view over 2700 pairwise comparisons yielded by 18 participants, represented by the means (dots) and standard errors (error bars) depicted in Figure 7.

The interpretation for parameter A (number of iterations) is straightforward by visual inspection: People prefer neural style transfer generated images with more iterations. Note-worthy is that the shape of the preference curve is linear only between 100 and 300 iterations, with a strong decline in the gradient thereafter. This suggests a convergence to a climax point near 500 iterations. The pattern looks different for parameter B (learning rate) that marks an inverted U-shape with the peak in the middle configuration corresponding to a learning rate of 2e1. Mentionable is the right skewness of the preference curve – this indicates that, between small (0.5e1 or 1e1) and large (4e1, 6e1) learning rates, people prefer neural style images created with smaller learning rates.

The preference curve for parameter C (content-to-style-weight ratio) shows an ambiguous picture with two peaks at the middle (100:100) and high configuration (300:100). This pattern cannot be interpreted as a U-shape with an anomaly peak because the latter levels with the middle configuration at par. This pattern rather reveals two different potential preference reasons – whereas the middle configuration allows to easily identify the person but in a strongly stylized form, the high configuration allows recognizing the person’s face subtleties better as the image appears closer to a photograph.

Discussion

The findings of this study allow a clear recommendation for configuring the neural style framework by Johnson (2016): Aesthetic results can be expected with 500 iterations, a learning rate of 2e1, and a content-to-style weight ratio of either 100:100 or 300:100. Interestingly, the preference curve for number of iterations reveals a distinct aesthetic preference (continuously positive and with small standard errors) even though the difference between the configurations was consciously hardly recognizable, as some participants remarked (see Figure 4).

Both optimal configurations for number of iterations and learning rate reveal a general preference for more realistic rendering because these configurations let the viewer best recognize the structure of the portrait image independent from the artistic stylization. There are several limitations to the findings of this study. The survey was tested by a small sample size. Even though this is consistent with other studies using the 2afc task, a higher sample size could allow detecting more differentiated preference patterns, e.g. differences between demographic groups. Moreover, the double-peak preference curve for the content-to-style-weight ratio could simply reflect a binary preference pattern between abstract versus realistic art. The survey did not control for this explanation. These aspects could be interesting focal points for future research.

The main contribution of this work is threefold. First, the 2afc methodology removes the measurement problems encountered with other measurement methods. Among them, the most common is the rating with Likert scales.
which is prone to acquiescence bias (Friborg, Martinussen, and Rosenvinge 2006). Rank ordering as an alternative
method is known to overwhelm the observer with a strong
cognitive demand to identify a relative ordering of all im-
gages (Palmer, Schloss, and Sammartino 2013).
Second, the findings provide designers with straightforward
recommendations on how they can use neural style transfer
as an effective design tool for creating artistic visual portrait
images. This may be a purpose of its own, or a means to
overcome their design fixation by producing artistic varia-
tions for inspiration.
Third, it reveals the existence of distinctly different aesthetic
preferences. It seems plausible to conjecture that the high
configuration for content-to-style weight ratio might be pre-
ferred by people with a focus on realistic depictions,
whereas the middle configuration might be preferred by
people who like abstract arts. This is consistent with other
research showing that people differ in their aesthetic prefer-
ences (Palmer, Schloss, and Sammartino 2013).
Taken together, under which conditions people favor the
proximity of the expected, i.e. a less artistic impression, or
a more stylized and thus abstract image rendition, is subject
to further research. The present work might have made a
crucial step towards opening this new research avenue.

Acknowledgment
This research was supported by the Yonsei University Fac-
ulty Research Fund of 2019-22-0199. The author would like
to thank Kyuha Jung for his support with the survey setup.

References
Champandard, Alex J. 2016. “Semantic Style Transfer and
Turning Two-Bit Doodles into Fine Artworks.” Arxiv.
Chen, Tian Qi, and Mark Schmidt. 2016. “Fast Patch-Based Style
Transfer of Arbitrary Style.” Arxiv.
Choi, Suk Kyoung. 2018. “Guess, Check and Fix: A
Phenomenology of Improvisation in ‘Neural’ Painting.”
Friborg, Oddgeir, Monica Martinussen, and Jan H. Rosenvinge.
2006. “Likert-Based vs. Semantic Differential-Based
Scorings of Positive Psychological Constructs: A
Psychometric Comparison of Two Versions of a Scale
Measuring Resilience.” Personality and Individual
Differences 40 (5): 873–84.
“Image Style Transfer Using Convolutional Neural
Networks.” The IEEE Conference on Computer Vision and
Pattern Recognition, 2414–23.
https://doi.org/10.1109/CVPR.2016.265.
Huang, Haozhi, Hao Wang, Wenhan Luo, Lin Ma, Wenhao Jiang,
Neural Style Transfer for Videos.” 2017 IEEE Conference
on Computer Vision and Pattern Recognition (CVPR),
7044–52. https://doi.org/10.1109/CVPR.2017.745.
Jiang, Shuhui, and Yun Fu. 2017. “Fashion Style Generator.” In
Proceedings of the Twenty-Sixth International Joint
Conference on Artificial Intelligence (IJCAI-17), 3721–27.
Johnson, Justin, Alexandre Alahi, and Li Fei-Fei. 2016.
“Perceptual Losses for Real-Time Style Transfer and
Super-Resolution.” Lecture Notes in Computer Science
(Including Subseries Lecture Notes in Artificial
Intelligence and Lecture Notes in Bioinformatics) 9906
LNCS: 694–711. https://doi.org/10.1007/978-3-319-
46754-6_43.
Leyssen, Mieke H.R., Sarah Linsen, Jonathan Sammartino, and
Composition in Multiobject Pictures.” I-Perception 3 (1):
Linsen, Sarah, Mieke H.R. Leyssen, Jonathan Sammartino, and
Size of Images of Real-World Objects.” Perception 40 (3):
Palmer, Stephen E., Karen B. Schloss, and Jonathan Sammartino.
Pelowski, Matthew, Gernot Gerger, Yasmine Chetouani, Patrick
The Classification of Images as ‘Art’ ‘Not Art’ and Correlation with Appraisal and Viewer Interpersonal
Differences.” Frontiers in Psychology 8 (OCT).
Pugach, Cameron, Helmut Leder, and Daniel J. Graham. 2017.
“How Stable Are Human Aesthetic Preferences across the
Lifespan?” Frontiers in Human Neuroscience 11 (May):
Psychological Review 34 (4d): 266–70.
Yang, Qian, and May L Ng. 2017. “Paired
Comparison/Directional Difference Test/2-Alternative
Forced Choice (2-AFC) Test, Simple Difference
Test/Same-Different Test.” In Discrimination Testing in
Zhu, Shizhan, Sanja Fidler, Raquel Urtasun, Dahua Lin, and Chen
with Structural Coherence.” ArXiv, 1680–88.
Exploring CC in XR: Visualizing Creative Conversation Topics to Facilitate Meaningful Face-to-Face Interaction

Hunter Harris, Makayla Thompson, Isaac Griffith, Paul Bodily
Department of Computer Science
Idaho State University
Pocatello, Idaho 83209 USA
harrhunt@isu.edu, thommaka@isu.edu, grifisaa@isu.edu, bodipaul@isu.edu

Abstract

With its ability to capture, model, and generate real-and-virtual combined environments, Extended Reality (XR) opens unique opportunities for designing creative agents. In this paper, we present HeyLo, a system for generating meaningful, virtual conversation cues in XR. HeyLo autonomously analyzes user tweets to identify common interests and visualizes these interests as conversation cues using emoji. We argue that this system demonstrates creativity using four metrics that characterize novelty, value, and intentionality in the domain of conversation cues: specificity, inter- and intra-user variance, and relevance. We discuss several potential research questions that we hope to answer in the future using this system and the broader implications of a creative system that is capable of bridging arbitrary interests to innovate its own creative ideas.

Source: github.com/harrhunt/HeyLo

Introduction

The field of Computational Creativity (CC) devotes itself to “the art, science, philosophy and engineering of computational systems which, by taking on particular responsibilities, exhibit behaviours that unbiased observers would deem to be creative” (Wiggins 2006). Much great work has been done to develop theory of CC, examining general questions such as: “What is creativity?” (Boden 2003); “What are computational frameworks and models for creativity?” (Wiggins 2006); “How do we measure creativity?” (Ritchie 2007); and “How do creative systems achieve autonomy?” (Jennings 2010). In tandem with these contributions, the field has fueled innovation of applied CC systems in domains as varied as culinary arts (Morris et al. 2012); linguistic constructs (Veale 2018); visual art (Colton 2012); poetry (Toivanen et al. 2012); narrative and story telling (Pérez y Pérez and Sharples 2004); mathematics (Pease, Gühe, and Smail 2010); software engineering (Colton, Powley, and Cook 2018); and Rube Goldberg machines (Xiou Ge and Varshney 2018). An important symbiotic relationship exists between these theoretical and applied contributions: the theory informs the application, and the applications inspire new inquiry and discussion of theory.

Because application is essential for the evolution of new theory, emphasis is rightly placed on encouraging research across and between an ever-widening spectrum of creative domains (Loughran and O’Neill 2017). It is one of the exciting challenges of the emergent field of CC to identify novel, valuable, and untapped domains in which to apply itself.

While there exist long-standing domains of creativity into which CC has yet to make an entrance, it occasionally happens that technology introduces an entirely new medium for creativity. This allows existing creative domains to take on new forms and provides computational systems novel access to new forms of creativity. An example of this is seen in the advent of extended reality (XR), an umbrella term that refers to all real-and-virtual combined environments (e.g., AR, MR, VR). Although XR dates back to the 1960s, commercially available and affordable XR to private consumers is a relatively recent phenomenon. This has prompted a significant increase in the demand for XR content (Moore 2017). In addition it has also opened new avenues of research and industry, including for example the use of XR to address issues of social isolation in older adults (Lin et al. 2018). Traditional production techniques have failed to meet this demand, leading to complaints about the lack of high-quality XR content. In response, some researchers have suggested that the solution lies in the development of procedural content generation (Tree and Malizia 2019).

The unmet demand for novel, creative XR content suggests unique and expansive opportunities for the field of computational creativity: opportunities for novel applications of CC, for evaluating how users interact with CC, and for identifying new lines of inquiry in CC theory. Although XR provides new opportunities for CC in familiar domains such as music, narrative, and visual art, of greater interest to the field is the question of “What forms of creativity have historically been inaccessible to computers to which XR provides unique access?” To explore this question, we set out to design a CC system for XR designed to leverage the unique purposes and strengths of this novel medium. The XR medium creates a space in which virtual and real worlds overlap to enhance real world interactions. It can capture, model, and influence interactions between people and organizations. How do these interactions relate to creativity?

Social psychology research demonstrates that face-to-face interaction produces up to 34 times more effective interactions than digital communication (Roghanizad and Bohns 2017). Research also demonstrates that a significant role in
the success of these interactions (from the viewpoint of an engaging party) is the degree of perceived similarity with other entities (Hampton, Fisher Boyd, and Sprecher 2019). Intuitively, this makes sense. We recognize the value of networking, building relationships of trust, and establishing common ground to improve the success of our interactions. Creativity is required to identify ways to establish common ground with others, and humans are not all equal at knowing what to say and how to say it. The “art of conversation” might essentially be defined as possessing the ability to engage in and maintain conversation about topics that are novel, valuable, and intentional—attributes that have been repeatedly used to define creativity (e.g., see (Ventura 2016; Wiggins 2006)). Can we conceive of a CC system that possesses this same ability?

We choose to focus on the simple task of generating meaningful conversation cues that effectively engage two parties in meaningful conversation. While this task can be done in mediums outside of XR (e.g., on social media), the task in the XR medium is unique: whereas on social media users largely interact with those who they intentionally seek out, in the real world people more frequently interact with those who they have not sought out or perhaps those who they might have otherwise intentionally avoided. The XR medium allows us to ask the question “Can a CC system be designed to suggest novel, valuable, and intentional conversation cues to engage two arbitrary parties in conversation that is meaningful to both parties?” We intentionally specify parties because the task we are describing is not unique to interactions between individuals. Whether it is two people meeting, two businesses interacting, a potential customer walking by a retail store, or a virtual chatbot, the problem is the same. Previous work has addressed this problem, but under the added assumption that interests are predefined and explicitly available (Nguyen et al. 2015; Jarusriboonchai et al. 2015). In our work, we consider the automated identification of interests to be an essential part of the problem to be solved. Our work is also unique in the visualization of interests using emoji.

In this paper we describe an extended reality computational creativity (XRCC) system, HeyLo, that operates in XR to identify potential topics of conversation between an XR user and another person encountered by the user in the XR medium and then overlaying visual representations of these interests using labeled emoji. To evaluate the creativity of the system, we define measures of novelty, value, and intentionality for artefacts generated by the system and apply these measures to comparatively evaluate the creativity of several different versions of HeyLo. We demonstrate examples of artefacts generated by HeyLo for real-world users. We discuss our future research agenda into XRCC and the implications of CC of a system that, more than merely specify interests, bridges seemingly-incompatible interests to propose novel concepts.

Methods
In this section we describe the design and operation of the HeyLo XRCC system. We first provide a high-level overview of the system. At the heart of this system is a model which attempts to identify interests from a set of user tweets that when used as topics for conversation evoke a sense of novelty, value, and intentionality. For purposes of illustrating different levels of creativity in solving this problem, we outline five different approaches for identifying interests. Finally we define four metrics by which we comparatively evaluate the performance of each of these five approaches.

HeyLo System Overview
The HeyLo system runs on an MR\(^1\) headset worn by a user with cloud support. Taking as input the image of a person from the headset camera, the system identifies a second user and then computes a set of weighted keywords representing shared interests between the two users. Each interest is paired with a representative emoji that the system overlays on the headset screen (see Figure 1).

To explicate where the creativity lies in this system and describe in further detail its implementation, we analyze HeyLo in terms of the FACE model (Colton, Pease, and Charnley 2011). In the FACE model, the creative behavior of a system is defined as a tuple of generative acts, containing 0 or 1 of eight possible types of creativity. Based on the behavior described in the system overview, we argue that the tuple for the HeyLo system is described by the tuple \(< F^9, A^0, C^9, E^9 >\). We consider the elements of this tuple in an order which best helps to describe the system and its creativity.

The Concept, \(C^9\) The concept of the artefact generated by the HeyLo system is a set of visual conversation cues. A visual conversation cue is defined as a text label representing a common interest between two users and an image (e.g., emoji) that represents the common interest. It is a set of cues that represents a concept because creativity is also required to ensure that cues within a set relate appropriately with each other (e.g., non-redundant, diverse, etc.).

The Expression of the Concept, \(E^9\) Figure 1 illustrates an expression of the concept expressed by HeyLo: a halo of labeled emoji around the face of a person. Of particular importance for the expression of visual conversation cues in HeyLo is that this expression occurs in the MR medium which allows the expression to take inspiration from and merge with the user’s normal field of vision. Expressing the concept in the MR medium allows the system to better achieve its intention: help the MR system user to identify effective means for starting a conversation in an arbitrary encounter without becoming a distraction to either user\(^2\).

The Method for Generating Expressions of a Concept, \(E^p\) HeyLo’s method for generating visual conversation cues hinges on generative models for three fundamentally creative tasks: first, identifying appropriate interests for a

---

\(^1\)Although HeyLo is specifically designed for MR (which gives the user access to their normal field of vision), the system can just as easily be used with AR or VR.

\(^2\)Current MR headsets would most certainly constitute an explicit distraction. With this statement, we are envisioning future MR devices that are as inconspicuous as glasses or contact lenses.
single user; second, identifying an appropriate method for finding common interests between users; and third, identifying appropriate images for visualizing the common interests that it generates. We use five different interest identification models and compare the results from each to find the best model for identifying a single user's interests. We will denote the interest identification model with $M$. The process below is the same for each model $M$ we use.

Details of the method which HeyLo uses to generate visual conversation cues are as follows. Given the user $u$ wearing the MR headset and an input image $f$ of a person taken from the headset camera,

1. Identify from the set of all users $U$ (i.e., all available social media users) the user $u$ represented in $f$ using facial recognition.\(^3\)

2. Compute a set of weighted interests for $u$ as follows:
   (a) Retrieve 1000 of the most recent social media posts created by $u$ and represent the content of these posts as a single multiset of words $S$. Filter $S$ for stopwords, URLs, and non-alphabetic characters, and for each word $s \in S$, replace $s$ with the NLTK (Bird, 2016) WordNetLemmatizer lemmatized form of $s$.
   (b) Apply interest identification model $M$ (described below) on $S$ to obtain a set of pairs $I_{M,u} = \{(k_1, w_1), \ldots, (k_i, w_i), \ldots, (k_n, w_n)\}$ where $k_i$ is a keyword or interest and $w_i \in \mathbb{R}_{\geq 0}$ represents a weight or level of interest for $k_i$. Normalize weights for interests in $I_{M,u}$ to range from 0 to 1 (i.e., divide each weight by largest weight in $I_{M,u}$).

3. Repeat step 2 to create $I_{M,u_{me}}$ for the MR headset wearer $u_{me}$.

4. Compute a set of shared, weighted interests $I_{M,\{u_{me},u\}}$ from $I_{M,u_{me}}$ and $I_{M,u}$ such that $I_{M,\{u_{me},u\}}$ is the set of all pairs $(k_i, w_i)$ where $k_i$ appears as an interest in both $I_{M,u_{me}}$ and $I_{M,u}$, and $w_i$ is the product of the two weights for $k_i$ in $I_{M,u_{me}}$ and $I_{M,u}$.

5. Reduce $I_{M,\{u_{me},u\}}$ to the pairs $(k_i, w_i)$ with the $l$ highest weights $w_i$.

6. For $(k_i, w_i) \in I_{M,\{u_{me},u\}}$ select an emoji $e_i \in E$ (where $E$ is the set of all emojis) as follows:
   (a) Use word2vec (Mikolov et al. 2013) to compute a vector representation $v_i$ for interest $k_i$.\(^4\)

---

\(^3\)This step is envisioned as part of future work and is not currently employed as part of HeyLo.

\(^4\)This step is envisioned as part of future work and is not currently employed as part of HeyLo.
(b) Identify an emoji $e_i$ from $v_i$ by computing
\[ e_i = \arg\min_{e \in E} d(v_i, v_e) \]
where $v_e$ is the vector obtained by applying word2vec to the text label for an emoji $e$ (where the label is multiple words, $v_i$ is the average of the vectors for each word), and $d$ is the function computing the cosine distance of two vectors (i.e., a measure of dissimilarity).

(c) Add the triple $(k_i, w_i, e_i)$ to the return set $R$.

7. Display the emoji $e_i$ with label $k_i$ for each triple $(k_i, w_i, e_i) \in R$ to form a halo around the face of $u$ in $f$ (e.g., see Figure 1).

An Aesthetic Measure, $A^G$ HeyLo possesses an aesthetic that favors interests that result from the overlap of the interests of two users. Furthermore, of the set of common interests, the system prefers those interests which are more heavily weighted by each user independently (as determined by the interest-extraction method). This is a temporary aesthetic until the system’s functionality has been expanded to be able to bridge non-overlapping interests. We plan to also incorporate the evaluation metrics of specificity, relevance, and inter- and intra-user variance (defined below) as an explicit part of the system’s aesthetic in future iterations.

An Item of Framing Information, $F^g$ HeyLo possesses a highly decomposed conceptualization model, which makes it easier for the system to describe its intentions and thinking in creating particular sets of conversation cues. It is not appropriate to provide this framing information in the MR medium (such would cause a distraction to the user), but providing the means by which the user can access this information is part of the fully-envisioned HeyLo system.

Evaluating Creativity in Conversation Cues

In developing HeyLo as a CC system, we discovered that several of the approaches we tried for identifying potentially creative conversation cues exhibited a lack of creativity. This was manifest in some of the approaches generating cues that failed the test of novelty, of value, or of intentionality. We found that this originated in decisions the system made for selecting an individual user’s interests. To objectively analyze how well-suited a particular approach is for identifying individual interests, we developed four metrics that collectively capture the notions of novelty, value, and intentionality for a set of conversation cues derived solely from the interests of a single user. These metrics are: specificity (value), intra-user variance (novelty), inter-user variance (novelty), and relevance (intentionality).

Specificity Finding common ground between two people is easy if that ground is poorly specified, however, such generality in finding common interests is unlikely to create a shared perception of similarity between users. To be valuable, a conversation cue must be specific. How can we quantify specificity? Consider that for two words $v$ and $w$, $v \ IsA w$ indicates that $v$ is more specific than $w$ (e.g., “field lacrosse” $\ IsA “sport”). We define in-degree($k$) for a keyword $k$ as the number of words $v$ such that $v \ IsA k$ is a valid relationship catalogued in ConceptNet. From this we define the specificity of an interest $k$ as
\[ \text{specificity}(k) = 1/(\text{in-degree}(k) + 1) \]
Note that if $k$ has an in-degree of 0 (e.g., as with $k = “field lacrosse”) $\ IsA “”, $k$ cannot be further categorized or specified. In this scenario, $k$ would receive the maximum specificity score of 1.0.

From this we can define the specificity of an interest identification model $M$. Let $I_M$ represent the set of unique interests extracted by $M$ across all users. Then the specificity of $M$ is the average of the specificity values for each unique interest:
\[ \text{specificity}(M) = \frac{\sum_{k \in I_M} \text{specificity}(k)}{|I_M|} \]

Intra-user variance Considering that HeyLo generates a set of visual conversation cues, the creativity of the system depends much on the diversity of cues in the set as it does on the cues themselves. In returning a set of interests representative of a user, a good model will extract a set of diverse interests. For a set of interests $I_{M,u}$ extracted by model $M$ for user $u$, we define the variance of $I_{M,u}$ as
\[ \text{variance}_{\text{intra}}(I_{M,u}) = \sum_{(k_i, w_i), (k_j, w_j) \in I_{M,u}} D_C(v_i, v_j) \]
where $D_C(v_i, v_j)$ is the cosine distance between two vectors $v_i$ and $v_j$ representing interests $k_i$ and $k_j$. Using the definition of intra-user variance for a set of interests, we define the intra-user variance of a model $M$ as the average of the intra-user variance values across all users:
\[ \text{variance}_{\text{intra}}(M) = \frac{\sum_{u \in U} \text{variance}_{\text{intra}}(I_{M,u})}{|U|} \]

Inter-user variance In addition to extracting a set of diverse interests, a good model will also extract diverse sets across users or, in other words, avoid repeatedly extracting the same interests for multiple users. To measure this inter-user variance, we find the sum of unique words for each user interest set divided by the total unique words across all user sets:
\[ \text{variance}_{\text{inter}}(M) = \frac{\sum_{u \in U} |I_{M,u}|}{\sum_{u \in U} |I_{M,u}|} \]

Relevance The success of an interest identification model depends on extracting interests that are not only specific and varied, but which also reflect the user’s actual interests. This final metric may be the most important of all, but is also one of the most challenging aspects to measure. A model’s predictions for a user’s interests can only be accurately assessed by the user him/herself. We are planning to conduct such a study as future work.

Interest Identification Models

The development of the HeyLo system included several iterations of testing of different interest identification models.
Table 1: Comparison of creative attributes exhibited by five interest identification models

<table>
<thead>
<tr>
<th>Metric</th>
<th>Empath</th>
<th>Retrained Empath</th>
<th>Raw Word Count</th>
<th>Bayesian</th>
<th>Chi-square</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specificity</td>
<td>0.008 ± 0.007</td>
<td>0.020 ± 0.076</td>
<td>0.092 ± 0.238</td>
<td>0.685 ± 0.374</td>
<td>0.275 ± 0.365</td>
</tr>
<tr>
<td>Intra-user variance</td>
<td>0.173 ± 0.057</td>
<td>0.141 ± 0.052</td>
<td>0.171 ± 0.072</td>
<td>0.115 ± 0.068</td>
<td>0.135 ± 0.083</td>
</tr>
<tr>
<td>Inter-user variance</td>
<td>0.046</td>
<td>0.090</td>
<td>0.230</td>
<td>0.999</td>
<td>0.828</td>
</tr>
</tbody>
</table>

Comparative Results

Comparative results can be found at https://www2.cose.isu.edu/~bodipaul/research/heylo/. Our goal in comparing different interest identification models was to identify which model produces artefacts exhibiting the most creativity. To evaluate specificity and variance, we calculated and averaged results for each approach over 500 of the most followed, publicly available Twitter handles. The results of these calculations are shown in Table 1. To evaluate relevance, we preselected five Twitter handles for five widely-recognized celebrities. We selected the celebrities from varying occupations and backgrounds to avoid returning similar interests for each example. We chose to perform the analysis on these users on the basis that their interests are generally well-known and therefore the results could be more easily assessed for how well the system’s intention of identifying user interests was achieved.

By the specificity and variance metrics the Bayesian model looks to have achieved a significant amount of novelty and (to some extent) value. Looking at Dave Ramsey’s results, for example, the Bayesian model extracts words that are specific and varied (e.g., delay, godly, backache, variable, and toolbox). These cues represent topics that most users do not often discuss. They also represent topics that are specific enough to avoid a conversation that is too general to carry meaning. These results, however, have very low relevance. Knowing that Dave Ramsey is a businessman, author, and renowned financial advisor, the interests extracted by the Chi-square model are significantly more relevant (e.g., money, advice, financial, debt, and millionaire). Inter-user variance scores that are at either extreme are undesirable because at the lower extreme (i.e., topics overlap) they lead to repetitive topics, and in the upper extreme (i.e., topics do not overlap) they lead to irrelevant topics. This leads us to conclude that inter-user variance scores that are not at either extreme are acceptable values for a given model. Both the default and retrained Empath models have very low inter-user variance scores reflecting that these models extract many of the same words across several users (low novelty). The Bayesian model has a very high inter-user variance score that leads to unique (high novelty) but irrelevant (low intention) topics. The raw word count and chi-square models both have intermediate values meaning they have an acceptable amount of inter-user variance to not suffer from the pitfalls of being at either extreme.

The low specificity score of the default and retrained Empath models betrays that these models also extract words which are relatively non-descriptive (e.g., play and party) and therefore likely to lead to conversations of low value. The raw word count model also suffers from low specificity (e.g., thank, thanks, and people). The Bayesian model has more specific words for each user; however the nature of this model leads it to prefer words that given the data are uniquely used by a particular user, even if these words are not the user’s interests. As the amount of data in the model increases that the Bayesian model will improve, but in its current state, this model suffers from over-specificity.

The results of the chi-square model exhibit keywords that are varied and specific but without being too specific (see Table 2). The NULL emoji for Nick Offerman’s word pawnee (a fictional city in a TV series featuring Offerman) was a result of the word pawnee not being in the Google News word2vec model. The model is unable to find the closest associated emoji because a vector for the word could not be determined. Future work will seek to address this issue.

---

1https://www.facebook.com/pages/category/  
2https://socialblade.com/twitter/top/500/followers
HeyLo Results using Chi-square Model

Our study allowed us to conclude that as a combination of relevance, specificity, intra-, and inter-user variance, the chi-square model was the best approach for identifying interests that would lead to creative conversation cues. Using this approach, HeyLo successfully finds meaningful topics of interest for two users and then identifies potential conversation cues from the overlap in these interests (see Table 3).

An example of where the system generated a good conversation cue is with the interest ukulele for Bill Gates and Nick Offerman. After looking into this more, Bill Gates sang while Warren Buffett played the ukulele in a video from 2016. Nick Offerman wrote a song on the ukulele and has even made his own ukuleles. A meaningful conversation between these two users might result from sharing their own experiences with ukuleles even though they seem to have very little in common.

Other results suggest potential areas of improvement in HeyLo. Some of the results such as great and thanks would not spark any meaningful conversations between users. To increase the quality of the conversation cues generated, a new method for finding connections between users is necessary. One solution to this problem is to creatively bridge seemingly disparate interests (discussed below).

Discussion and Conclusion

In developing our vision of an XRCC system for visualizing conversation cues, we established several milestones. HeyLo, as presented in this paper, accomplishes the first of these milestones, which is to design an XRCC framework that autonomously elicits user interests and visualizes conversation cues based on common interests between users with the intention of facilitating meaningful interaction between them. The current system may yet benefit from a refinement of the metrics used to measure different interest identification models (e.g., leveraging work done by Joho and Sanderson (2007)). An outline of our next steps are outlined in the following research questions:

1. Can HeyLo effect meaningful conversation through the proposal of visual conversation cues based on distinct yet compatible interests (e.g., Switzerland and chocolate)?
2. Can HeyLo propose visual conversation cues through the creation of novel interests that form from bridging disparate interests (e.g., computer science and public defense)?
3. Can HeyLo’s intention be augmented to account for the polarity (+/-) of a user’s sentiment towards an interest?
4. Can HeyLo’s intention be augmented to account for an individual’s mood (e.g., based on facial expressions)?
5. Can HeyLo’s creativity be used to suggest pairs of users who are likely to engage in meaningful conversation?

We will focus on questions 2 and 3 from the list above and discuss their significance to our system’s expansion.

Bridging Seemingly Disparate Interests If common interests cannot be found, one solution is to redefine the space of user interests through the creation of novel user interests that would be common to both users. As an example, consider the following social encounter between one of the authors (a computer science professor) and a neighbor (a public defender). The meaningfulness of this particular interaction was initially stifled by an apparent lack of common interests between the author and the neighbor. The meaningfulness of the conversation dramatically increased, however, when the author began to seek for ideas for potentially bridging the two parties’ disparate sets of interests and came up with the idea of teaching computer science to indigent defendants. This idea eventually became the impetus for the

<table>
<thead>
<tr>
<th>User</th>
<th>money</th>
<th>advice</th>
<th>financial</th>
<th>debt</th>
<th>millionaire</th>
</tr>
</thead>
<tbody>
<tr>
<td>@DaveRamsey</td>
<td>money-bag.png</td>
<td>warning.png</td>
<td>bank.png</td>
<td>credit-card.png</td>
<td>man-farmer.png</td>
</tr>
<tr>
<td>@realDonaldTrump</td>
<td>transcript</td>
<td>witch</td>
<td>impeachment</td>
<td>democrat</td>
<td>shifty</td>
</tr>
<tr>
<td>@tonyhawk</td>
<td>ice-skate.png</td>
<td>snowman.png</td>
<td>owl.png</td>
<td>building-construction.png</td>
<td>skateboard.png</td>
</tr>
<tr>
<td>@Nick.Offerman</td>
<td>man-bouncing-ball.png</td>
<td>cat-with-tears-of-joy.png</td>
<td>cherries.png</td>
<td>man-artist.png</td>
<td>NULL</td>
</tr>
<tr>
<td>@BillGates</td>
<td>mosquito.png</td>
<td>cancer.png</td>
<td>hourglass-not-done.png</td>
<td>children-crossing.png</td>
<td>pill.png</td>
</tr>
</tbody>
</table>

Table 2: Visualized conversation cues generated by HeyLo based on individual user interests
creation of a new introductory computer science course in the local women’s correctional center.

The process of taking known ideas and finding novel, valuable, and intentional bridges between them that lead us to new findings or applications of knowledge might be considered the essence of creativity itself. Our goal is to develop HeyLo into a system that explicitly models this process by bridging seemingly disparate interests in unique ways.

The ability to bridge seemingly disparate interests has significant applications for business and consumers. Bridging the dissimilarities between the consumer’s interests and the business’ offerings allows both the consumer and the business to be more efficient in their interactions. Consider as an example a user who is interested in computers and programming walks by a clothing store, but has no interest in clothing. The system could bridge the dissimilarity and suggest a pair of compression gloves that help with carpal tunnel syndrome. The user now has the opportunity to have a successful interaction with a business they otherwise would not have had. The challenge of bridging topics has been the subject of significant research (e.g., (Berthold 2012; Olsson et al. 2020; Nguyen et al. 2015)) which we plan to incorporate into future work.

**Polarizing Interests** We define a polarized interest as an interest together with an individual’s sentiment toward that interest. Currently HeyLo disregards the polarity of a user’s sentiment towards their identified interests. Incorporating polarity, however, gives the system an increased capacity for intentionality. There are many scenarios in which the system can use the polarity of interests to better achieve the intentions of the user. Consider the following examples:

- A user only wants to see common interests for which both users share the same polarity.
- A user wants polarity visualized so that they can be alerted to an individual’s sentiment toward a particular topic (e.g., to approach controversial topics tactfully).
- A user wants to filter the interests they see based on polarity. This can go two ways: the user only wants to see interests towards which an individual feels favorably; or the user only wants to see interests towards which an individual feels unfavorably (e.g., for purposes of engaging with different points of view or for sparking debate).

From the examples above, there are many different intentions for social interaction that can be derived from using polarized interests. It is important to note that not all of the interactions described above are positive in their intentions, opening the need for ethical considerations in expanding HeyLo in this direction.

We have suggested that XR presents novel research opportunities for CC. As an example, we have presented HeyLo, an XRCC designed to autonomously generate visual conversation cues when encountering other users in XR environments. We have discussed measurements of specificity, variety, and relevance as means of evaluating novelty, value, and intentionality in this domain and demonstrated a comparative analysis of variations of the HeyLo
system using these metrics. The system we have presented represents a basic framework for XRCC in which we hope to continue research into how to effectively find bridges between seemingly disparate interests in order to generate creative visual conversation cues.

References


Interactive Neural Style Transfer with Artists

Thomas Kerdreux∗
Department of Computer Science
ENS, PSL University & INRIA
Paris, France

Louis Thiry∗
Department of Computer Science
ENS, CNRS, PSL University
Paris, France

Erwan Kerdreux
Department of Design
ENS Paris-Saclay
Palaiseau, France

Abstract
We present interactive painting processes in which a painter and various neural style transfer algorithms interact on a real canvas. Understanding what these algorithms’ outputs achieve is then paramount to describe the creative agency in our interactive experiments. We gather a set of paired painting-photograph images and present a new evaluation methodology based on the predictivity of neural style transfer algorithms. We point some algorithms’ instabilities and show that they can be used to enlarge the diversity and pleasing oddity of the images synthesized by the numerous existing neural style transfer algorithms. This diversity of images was perceived as a source of inspiration for human painters, portraying the machine as a computational catalyst.

Introduction
Neural style transfer (Gatys, Ecker, and Bethge, 2015), which seeks at rendering the content of one image using the style of another, provides impressive results as it takes advantage of the rich hierarchical representation of images produced by convolutional neural networks (CNN) to quantify the style and content of images. The many ways to manipulate these complex maps, as well as their increasing ease of implementation, have hence underpinned the development of a plethora of successful methods in this area of computational artistic rendering.

Several evaluation techniques exist to compare all different methods. On the one hand, many methods focus on quantifying how much a neural style transfer method attains a numerical objective. These are good engineering indicators, but we highlight that they are not necessarily relevant at measuring the quality of the outputs of style transfer algorithms. On the other hand, qualitative evaluation methods typically consist of collecting a large number of subjective impressions on the algorithms’ outputs. It provides average scores on the content preservation and style quality of the algorithms’ outputs. However, it does not reveal the specificity of an algorithm compared to another.

To have a more precise characterization of these algorithms, we introduce a new evaluation methodology based on the predictivity of neural style transfer algorithms and gather a set of paired painting-photograph for this evaluation. The predictivity consists of assessing whether or not the algorithms’ outputs are close to an existing painting when using this painting as the style image and the associated photograph as the content image. This is also a crucial point in the computational creativity perspective as some outputs are deemed interesting while bearing not much resemblance with the initial painting, i.e. with what the painter did.

Besides, when showing artists some outputs of style transfer algorithms using their paintings as style images, they often do not observe their practice. However, they sometimes identify inspiring aspects in the various outputs of different algorithms, implicitly acknowledging their computational creativity. This naturally led us to painting processes with artists, who could not only edit groups of style transfer outputs but use them as basic elements to widen their style. This constructively interlaces the agency attribution of the algorithms part in the creative process.

We further encouraged this complexity by exploring these algorithms in the real world, where the outputs are projected onto a real canvas, the classical space for human painters. Human and machine contributions are then mingled in a single canvas. Interestingly then, to help the observer that seeks to untangle the contribution of each agent, the canvas can be shown together with the various computational suggestions. In such creative processes, the algorithms were experienced as computational catalysts to human creativity, a middle ground between agents that are creative on their own and still technical tools.

Outline. We first describe the new methodology for qualitative evaluation of different style transfer algorithms. We then question the relevance of existing quantitative evaluations giving a simple example where improving the quantitative criterion of an algorithm does not improve the quality of the stylization quality of the outputs. We then indicate that some approaches to neural style transfer do not satisfy a basic property, which leads to an instability behaviour that ultimately allows reinforcing the diversity of style transfer outputs.

Based on all these observations, we present various interactive painting experiments between human and style transfer outputs. This leads us to the notion of computational catalysts that help to characterize the algorithms’ contribution in our specific settings.
Evaluating Neural Style Transfer Methods

Neural style transfer methods have been proposed to transfer the style of any painting while preserving the content. A popular class of such algorithms is based on convolutional neural networks (CNN). These methods are particularly useful for style transfer, which is the process of transferring the style of one image to another.

Quantitative evaluation methods for style transfer have been proposed. Some of these methods measure the content preservation and the stylization quality of the output. Others are based on CNN features and are much faster than the optimization-based approaches, but do not say in what sense the style and content are better preserved.

To have a systematic and more refined comparison, we propose to study the predictivity of style transfer algorithm: does an algorithm stylize the image in a way similar to what a painter would have done? Precisely, when considering a photograph as a content image and a figurative painting of this image as a style image, one can compare the output of the neural style transfer algorithm with the figurative painting and further judge whether the style transfer technique succeeds in predicting the painting, and if not, try to characterize how it differs from it.

Such pairs of photographs and content-preserving paintings are not readily available; landscapes are constantly changing, face portraits are rarely faithful to the original and we rarely possess the photograph of the model. Building paintings, however, is a good class of paintings for the proposed study. We thus construct a set of photographic-painting pairs, see Figure 2 for instance, focusing on the Series Notre Dame de Rouen Cathedral by Claude Monet. It consists in about forty paintings capturing the facade of Notre Dame de Rouen Cathedral from nearly the same viewpoint at different times of the day and year and under different meteorological and lighting conditions (Kleiner, 2009, p. 656).

With this set, qualitative evaluation can be done more systematically and less arbitrarily: in the example shown in Figure 2, STROTSS output is qualitatively the closest to the Monet painting, especially for the lightening effect on the door and the left of the portal. Gatys and WCT suffer from spatial inconsistency as the blue sky is replaced by a sunlight halo in the first one and the background is hardly distinguishable in the second one. We release this set together with the outputs of the style transfer algorithms to facilitate and systematize the qualitative evaluation of neural style transfer techniques.

Quantitative evaluation

Numerical evaluation methods have the benefit of being more systematic and objective. However, we point here that most neural style transfer evaluation methods are specific to certain algorithms and are not always relevant for the stylization quality of the output.

In computer vision, perceptual losses are becoming the standard to compare the visual similarity between images (Zhang et al., 2018). These methods based on CNN features...
comparison offer state of the art performance on image similarity judgment datasets using different CNN architectures. Since neural style transfer originally consists of optimizing an image in order to match the CNN features of another style image, the perceptual loss between the outputs and the target style image might be artificially small despite notable perceptual differences.

Other numerical evaluation techniques were proposed; Sanakoyeu et al. (2018) test whether a pre-trained neural network for artist classification on real paintings succeeds in classifying the artist of the style image based on an algorithm’s output. Jing et al. (2017) consider comparing saliency maps between images since the spatial integrity and coherence of the saliency maps should remain similar after style transfer. Moreover, as neural style transfer relies on a certain quantification of the style based on CNN features, Jing et al. (2017) propose to evaluate how much the optimization objective is achieved in style transfer. We show in the following case that improving the optimization objective is not necessarily related to the visual quality of the output.

Optimization-based neural style transfer methods consist in optimizing the pixels of an image \( I \) to minimize a loss \( l \). This loss \( l \) is usually the sum of a content loss \( l_c(I, I_c) \) measuring the content similarity between \( I \) and the content image \( I_c \), and a style loss \( l_s(I, I_s) \) measuring the style similarity between \( I \) and the style image \( I_s \). In the STROTSS method, Kolkin, Salavon, and Shakhnarovich (2019) define the style loss as the Earth Movers Distance (EMD) between CNN features of the image \( I \) and the style image \( I_s \). Given the CNN features \( \Phi(I), \Phi(I_s) \) of the images \( I, I_s \), we compute the distance matrix \( C^{I,I_s} \) between \( \Phi(I) \) and \( \Phi(I_s) \) and the EMD is defined as the solution of the following optimization problem

\[
EMD(I, I_s) = \min_{T \geq 0} \sum_{i,j} T_{ij} C_{ij}^{I,I_s} \text{ subject to } \sum_{i} T_{ij} = 1/m \quad \text{ and } \quad \sum_{j} T_{ij} = 1/n.
\]

Exact EMD computations are too expensive for neural style transfer applications, and a relaxed EMD (REMD) is used in STROTSS. It consists in taking the maximum of two simple lower bounds of the EMD, each obtained removing one of the two linear constraints sets \( \sum_{i\neq j} T_{ij} \) applied on the transport plan \( T \)

\[
\text{REMD}(I, I_s) = \max \left( \min_{T \geq 0, \sum_{i} T_{ij} = 1/m} \sum_{i,j} T_{ij} C_{ij}^{I,I_s}, \min_{T \geq 0, \sum_{j} T_{ij} = 1/n} \sum_{i,j} T_{ij} C_{ij}^{I,I_s} \right).
\]

Despite the use of this loose relaxation, the human evaluation done via Amazon Mechanical Turk (AMT) indicates that STROTSS statistically offers the best style/content trade-off compared to the other neural style transfer techniques (Kolkin, Salavon, and Shakhnarovich, 2019, §4) in the opinion of the AMT workers. Experiments done with artists confirmed this trend as the artists were mostly impressed by results produced by STROTSS.

The authors mentioned that a better approximation may yield better style transfer results. Sinkhorn-distance (Cuturi, 2013) is a good candidate to this purpose. We thus replaced the relaxed earth movers distance REMD by the Sinkhorn earth movers distance

\[
\text{SEMD}_\epsilon(I, I_s) = \min_{T \geq 0} \sum_{i,j} T_{ij} C_{ij}^{I,I_s} + \epsilon \sum_{i,j} T_{ij} \log T_{ij} \text{ subject to } \sum_{i} T_{ij} = 1/m \quad \text{ and } \quad \sum_{j} T_{ij} = 1/n,
\]

where \( \epsilon \) is the entropic regularization parameter. The corresponding optimization problem is convex and is solved iteratively with a fixed number of iterations \( N \). SEMD\(_\epsilon\) is an upper-bound of the EMD and it converges to the exact EMD as \( \epsilon \) goes to 0. We release a Pytorch (Paszke et al., 2019) implementation\(^2\) of STROTSS including the SEMD.

Figure 3 shows a comparison of experimental results, suggesting that getting much closer to the mathematical quantification of the style does not necessarily lead to more relevant results, and numerical evaluation of how much the mathematical objective is achieved is not essential from a visual perspective.

In the same idea, the instability phenomena that are commonly assumed to be detrimental in the neural networks literature (e.g. adversarial examples), can qualitatively increase the creative possibilities of neural style transfer.

**Instability phenomena**

Neural style transfer instabilities have been pointed out by Rissier, Wilmot, and Barnes (2017) and Gupta et al. (2017) in the case of real-time style transfer for videos. The aim is to identify and remove the time-inconsistent artefacts that create unpleasing effects. Here we outline instabilities stemming from another type of inconsistency and propose to take advantage of them.

A style transfer method is simply a function \( f \) that takes as input a style image \( s \) and a content image \( c \) and outputs a stylized version \( f(s, c) \) of \( c \) with \( s \). It is reasonable when giving such a method the same image as content and style, to expect the image itself, i.e., that \( f \) satisfies \( f(s, s) \approx s \). Let us now consider the following recursion

\[
x_{t+1} = f(x_t, x_t),
\]

\(^2\)https://github.com/human-aimachine-art/pytorch-STROTSS-improved
where $x_0$ is an initial image. Optimization based methods empirically converge to an equilibrium where $f(x, x) = x$ independently of the initialization. On the opposite, feed-forward approaches to style transfer (Ulyanov, 2016; Johnson, Alahi, and Fei-Fei, 2016; Li et al., 2017; Huang and Belongie, 2017) lead to oscillating sequences $(x_t)$ around non-trivial (i.e. not a monochrome image) forms, yet typically bearing absolutely no resemblance with the initial image $x_0$. Since the pixel values are clamped between 0 and 1, colours end up being either saturated or zero, but not uniformly and still revealing specific patterns in Figure 4 for instance. Interestingly also, when starting from very simple images $x_0$, like a uniform color, for some $f$, the sequence would still show the same type of instability in the long-run, see this video\(^3\) for instance.

From the perspective of computational creativity, this apparent failure is interesting. In the first iterations, we observe that some methods produce a series of images progressively stylized. Given a style transfer function $f$, the very same effect happens across all sequences we experimented with. For instance, in Figure 6 we see a distinct tessellation effect in the images of the first row. We use this technique in the interactive painting experiments to produce more diverse and computationally creative style transfer outputs, see image (f) in Figure 9 for instance.

Alternatively the asymptotical regime of the sequences $(x_t)$ produces surprising animations. The appearing patterns are completely different from one approach to another, but are experimentally the same for different initialisation images and a given method. Sequences are shown in Figure 5, refer to this video\(^4\) or this one\(^5\) for a more lively visualisation.

**Interactive Painting Experiments**

In the previous sections, we have questioned the relevance of neural style transfer evaluation. To go beyond comparing techniques, we propose to take advantage of the diversity of the outputs and to use them as a source of inspiration for artists.

Some painters have recently explored interactive processes with machines in the real world, particularly in the case of painting. For instance, Chung (2015) among others, leveraged on the algorithms from artificial intelligence to paint interactively with humans in the real world, where a machine would act on the real canvas via a robotic arm. Cabannes et al. (2019) also explore such an interaction, where the machine does not act but suggests via projection. However, none of these use style transfer algorithms outputs to paint interactively with an artist on the canvas.

We explore that possibility through various series of interactively painted portraits. We describe here various interactive painting experiments inserting outputs of neural style transfer algorithms during the human painting process. In all cases, the algorithms’ creations are projected onto the

---

\(^3\)https://youtu.be/WCJNLWb-H2M

\(^4\)https://youtu.be/gAq1lvb1G1c

\(^5\)https://youtu.be/s87R-9JITvE
Figure 6: Each row represents the first fourth iterate of sequence defined in Equation (1) with a different neural style transfer approach. The first row corresponds to MST (Zhang et al., 2019b), the second to WCT (Li et al., 2017) and the third to AdaIn (Huang and Belongie, 2017). The first method has a clear tessellation effect, the second has a blurring effect and the third gives an increasingly clownish style to the image. These effects appear to be the same across images, see Figure 4.

canvas but never automatically painted, for instance via a robotic arm or a printer. We first describe the experiments on canvas and then motivate the underlying design choices. We finally show how the notion of computational catalyst naturally emerges. Note also that all the paintings revolve around portrait themes.

The portraits

Editing multiple styles in one portrait. Neural style transfer outputs are very diverse from one method to another, as outlined in Figure 1 for instance. To edit the creative content of these outputs into a single final artwork, we select a person’s photography as a content image and we transfer the style of previous artists’ painting into this content image using various algorithms. We then show the stylized images to the artist, but not the original content image, which he never sees. He then selects some of the outputs that best resonate with his practice. These style transfer outputs are finally alternatively projected on a canvas for a certain amount of time. Figure 7 is an example of canvas realized according to this process; we complement the final canvas with the various style transfer outputs chosen by the artist. We also explore other variations of that idea, for instance via collage, where the selected outputs are mingled together into a single image which is projected afterwards.

Pixelizing portrait construction. The motivation of this creative process is to artificially create an interactive loop between the painter and the algorithm. The initial image is projected onto or next to the canvas, which is divided into squares. The painter is then asked to paint sequentially on each square of the canvas. Whenever a square is completed, we use it as a new style image to stylize the initial photographic-like image of the portrait, which is then projected on the canvas, images (a)-(d) in Figure 8 are some of these projected outputs. Anytime then, the painter only sees an interpretation of the photographic portrait by a style transfer algorithm taking the painter’s style in the previous painting. We show one example of a canvas produced in such a way in Figure 8.

Note that the style transfer output should be the machine’s prediction of what the artist would do, provided at least that the previous square contains all the style information of the painter and that the style transfer method is ideal. This remark was then the basis for a gamification exploration of the painting process, where the artist was asked to attempt to fail the machine prediction as much as possible.

This decomposition of the painting process produces painting artworks that are sequential objects. All of the successive iterations of the painting are of interest, and not only on the final canvas. Figure 7: Left: two panels of 100×50 cm oil canvas. Right: the style transfer outputs that formed the painter theme and were successively projected on the canvas. For each, the three canvas in Figure 10 were used as style images; the content was a photographic portrait.

Figure 8: (Top) Final canvas and steps 1, 5, 7. (Bottom) Original photographic and projections after steps 1, 5, 7. For this 50×50 cm oil pastel canvas, the blank canvas was divided into 9 squares and the painter had paint sequentially on each square. After each square was completed, the output of the style transfer algorithm using the original photograph as a content and the current canvas as a style image was projected.
the final canvas. Actually, algorithms in image computational creativity are very much less performative at generating images as sequences of brushstrokes as they are at generating images all at once, like with GANs (Goodfellow et al., 2014) for instance. This is simply because paintings are usually not sequential objects. Indeed we very rarely observe all the steps leading to a painting, apart for large quantities and categories of simple sketches (Eitz, Hays, and Alexa, 2012; Jongejan et al., 2018). Alternatively, computationally inferring the steps of a painting from the final canvas (Xie, Hachiya, and Sugiyama, 2013; Ganin et al., 2018; Nakano, 2019) is not yet very successful. This arguably explained why in painting art, compared to other domains such as music, whose artworks are sequential by nature, the computationally creative algorithms are harder to frame in a fully interactive way with humans, hence limiting the ability for a painter to truly interact with machines.

Interactive series of portraits. We then considered using neural style transfer outputs for series of portraits. We select a photographic portrait and we stylize with a neural style transfer algorithm it using a previous artists’ painting as a style image. We project the stylized image as inspiration for the painter. When the painter has finished the painting, we stylize again the photographic portrait using the painting that has just been painted. We project the new stylized image as the next inspiration, and we repeat the process, typically two or three times. Figure 9-10 present two series of canvas in chronological order. In figure 9 all canvas stem from the same photographic portrait, while in Figure 10 we alternate between two photographic portraits to avoid specialization of the artist to particular content.

We played on the many ways to generate new style images at each iteration. Importantly, at each iteration, the previous image serves as a style image. This allows for the artist to interact with a computational version of his past work, a key aspect computational creativity has to offer.

Also in Figure 9, the photographic portrait is an input in the first canvas. The subsequent machine only uses as content or style image the preceding paintings. The observed divergence is hence an intertwined responsibility between the painter and the algorithms.

Discussion

Neural style transfer algorithms are computationally creative in the sense that they may produce new images with an aesthetic that can significantly differ from what a painter would do. In order to turn this creativity into artworks, we have specified various painting experiments on a real canvas between a painter and outputs from these algorithms. We now report how these attempts shed light on a few aspects of the computational creativity of neural style transfer algorithms and cast them, in this specific setting, as computational catalysts to human creativity. Besides, the interactive painting process itself was designed to embody some questions related to computational creativity and to human-machine interplay, which has arguably become a major societal theme.

Computational Creativity and Catalyst. The initial motivation for designing human-neural-style-transfer interac-

![Figure 9: (Top) First, second and third canvas. (Bottom) First, second and third projections. Three 50×50 cm oil on canvas of the same face. Iteratively showing style reinterpretation to the painter. Image (a) served as a style image to produce (e); Image (b) served as a style image to produce (f). Note that outputs (f) is the third iterate of Equation (1) with the MST style transfer algorithm, in order to produce a slight tessellation effect.](image1)

![Figure 10: (Top) First, second and third canvas. (Bottom) First, second and third neural style transfer outputs. The top row collects pictures of the paintings, 50×50 cm oil on canvas. The bottom row gathers the outputs of the neural style transfer (NST) methods which were chosen by the painter as his basis theme. The first is the stylization of the photographic image with a previous painting of the painter. The second one is the stylization of a different photographic image with the first painting as a style image. The third is the stylization of the same previous photographic image with the second painting as a style image. Note that the order of the painting is chronological.](image2)
tive experiments was to create a single object out of many different style transfer outputs, focusing here on a painting instead of a printed version of the numerical output. This echoes other creative works with machines where some artists playfully worded themselves as editors of the machine creativity, see for instance the rationale surrounding the last A.I. assisted musical album Chain Tripper of the band YACHT (2019). Though during our painting experiments, the intertwining between the machine’s outputs and painter interpretation was non-trivial since the painter was altering the machine suggestions. The painter felt the outputs were giving new style directions, wording them as computational catalysts to his own creativity.

In these interactions with algorithms, we exploit the ability of style transfer methods to produce outputs based on the previous works of the painter. This is a simple yet powerful idea that allows an artist to interact with computationally influenced versions of its own (past) work. This was felt by the painter as a semi-extraneous interpretation of his past techniques, allowing him to rediscover some elements of his old practice in a surprising way. Besides our specific framework, this seems to be another major benefit and specificity of computational creativity.

Importantly also in these portrait paintings, the artist could not see, except in the beginning, the real photographic portrait. We purposely designed it in this way so that the painting practice could embody the fact of perceiving the world only through the machines’ lens. This has critical societal echoes; for instance, the issues raised by the so-called fake news stems to some extent from generative algorithms capacities, a technical point of view. But it concomitantly, the societal point of view, comes from our increasingly resorting to numerical pieces of information as a way to perceive the world. Here we hence implicitly explore what a painter felt when relying only on machine outputs to see the portraits.

Alternatively, it also gives another perspective on computationally creative algorithms, as offering new inspirational spaces to portray. Indeed we may not only explore algorithms outputs through printed versions, pretty much as we do not capture nature only through photography. Computationally creative outputs may hence be thought of as new types of landscapes for painters to capture.

Note also that the transient essence of these computational landscapes has very different rules than that of Nature; by erasing the content files or algorithms outputs, the painting could remain the only imprint of the machine outputs. This again is a specificity of computational creativity, when framed as a theme creator for artists, that is worth exploring.

**Designing Human-Machine painting processes.** A major aspect in these human-machine interactive processes is that we engineered the numerical outputs out in the real world, rather than having a painter to interact with machines on a tablet for instance.

Indeed when the painting process materializes in a numerical tablet, it strongly constrains the painter’s sensations; he does not feel the brushstrokes’ gesture, the canvas is not perceived in the full-dimensional space, etc. Even with interactive experiments on a real-canvas, the painter felt some processes as being too intrusive or constraining, like the experiment reported in Figure 9 which forces the artist to follow unusual rules for creating. This highlights that computational creativity, when considered in such a human-machine interplay, is notably conditioned by the current state in the engineering of such interactive systems. For instance how much projection is less intrusive than a robotic arm?

This level of machine’s intrusion is inherently linked with how the computational creativity of the algorithms is perceived, notably concerning the creative agency that is attributed to the machine outputs. Part of the discussion around computational creativity may hence be tightly related to some artists’ feelings of losing a share of the creative agency when algorithms become more than a disposable tool.

So it appears that when engineering such systems, there are typically two directions in the interfacing, either the machine goes out of the numerical world or reversely, the human interacts with the machine in the numerical world. And as we described previously, the interfacing puts the human artists in very different situations. However, this may not only be considered as a limitation as each constraint forces the painter to embody what we may feel in our daily interaction with machines. In particular then, each type of interfacing echoes, and may advocate then, a different societal relation humans have with machines; at the era of machines, it is primal to explore many such experiments.

**Plastic point of view.** These interactive painting experiments were also designed to explore pictorial aspects.

For instance the photographic portrait that initiate the series in Figure 10-9 were in black and white. However, the style transfer algorithm and the painter were not constrained in the grey-scale space. The painter could observe in projected outputs of the machine, or conversely initiate in the real canvas, the emergence of the colours. For instance, in Figure 10, red appears on the eyebrows while in Figure 9 the colours are intended as variations of shade, which only exists through the machine.

While this is interesting from the creation point of view, it is also from the observer who is concerned about agency attribution. For a given aspect of the painting, like colours, did the painter simply repeat the machine colourization outputs, re-interpreted it or even started it? This reinforces the importance, in an exhibition, of algorithms’ outputs as testimonies of the final artworks.

**Conclusion**

We present interactive painting experiments between neural style transfer outputs and a painter. It reveals many potential benefits of leveraging computational creativity in this type of interactive framework and questions some computation aspects of neural style transfer specifically.

**Acknowledgments** The authors would like to thanks Tomas Angles, Thibault Séjourné and Stephane Mallat for interesting and helpful discussions as well as Vivien Cabannes and John Zarka for rereading.
References


Mroueh, Y. 2019. Wasserstein style transfer.


Let’s Figure This Out: A Roadmap for Visual Conceptual Blending

João M. Cunha, Pedro Martins and Penousal Machado
CISUC, Department of Informatics Engineering
University of Coimbra
{jmacunha,pjmm,machado}@dei.uc.pt

Abstract
The computational generation of visual representation of concepts is a topic that has deserved attention in Computational Creativity. One technique that is often used is visual blending – using two images to produce a third. However, visual blending on its own does not necessarily have a strong conceptual grounding. In this paper, we propose that visual conceptual blending be used for concept representation – a visual blend complemented by conceptual layer developed through elaboration. We outline a model for visual conceptual blending that can be instantiated in a computational system.

Introduction
It is often said that an image is worth more than a thousand words. Such is aligned with the views from Petridis and Chilton (2019) who state that visual advertisements – specifically when visual metaphors are used – are much more persuasive than plain advertisements or text alone when conveying a message.

Visual metaphors result from a process of visual blending grounded on a conceptual level. The technique of Visual Blending (VB) consists in merging two or more visual representations (e.g. images) to produce new ones. On the other hand, Conceptual Blending consists in integrating two or more mental spaces – knowledge structures – in order to produce a new one, the blend(ed) space (Fauconnier, 1994; Fauconnier and Turner, 2002). When CB and VB are used together, the process can be referred to as Visual Conceptual Blending (Cunha, Martins, and Machado, 2018a), which we consider to have an important role in the production of Visual metaphors.

Visual metaphors are sometimes hard to decode by humans – Petridis and Chilton (2019) report 41.3% accuracy in correctly interpreting visual metaphors. Getting computational systems to understand them is even harder – most of the success comes not from analysing images but from using the accompanying text (Petridis and Chilton, 2019). On the other hand, there is promising work regarding computational approaches to the production of Visual Metaphors.

Cunha and Cardoso (2019) highlight the importance of having a conceptual ground for producing visual blends, creating a connection between conceptual blending and visual blending. This idea was already referred by several authors but, as far as we know, no concrete computational model has been proposed.

Existing research that relates to the proposal of a visual conceptual blending model somehow comes short of achieving such goal. Karimi et al. (2018) initially present their work as a computational model for generating visual conceptual blends in the domain of sketching. However, the core of the model is more related to conceptual shifts – retrieving sketches similar to an initial one – than with visual blending, which is later presented as a possible application and not intended as an automatic process.

Chilton, Petridis, and Agrawala (2019) propose a workflow for producing Visual Blends, which allows users to generate visual blends collaboratively. The workflow is composed of three main steps: brainstorming, synthesis, and iteration. The user is responsible for finding suitable related concepts (association), retrieving appropriate images that represent the gathered concepts and annotating them in regards to the shape of their elements. The system then finds matches based on shape between the annotated images and combines them into a blend. The user is then responsible for evaluating the results. The process can be repeated until the...
users are satisfied. We consider this system as more close to a creativity support tool than a creative system.

Xiao and Linkola (2015) propose a workflow for generating visual compositions to express certain meanings, composed of three tasks: (i) finding photos of the subject and message; (ii) preprocessing photos; and (iii) applying visual operations (juxtaposition, replacement and fusion) to produce blends (see Fig. 1).

Other systems exist in which the conceptual layer can be considered reduced as they often rely on a mere mapping between the input concepts and the visual representations used in the blend (Cunha, Martins, and Machado, 2018a; Zhao et al., 2020). Nonetheless, the system by Cunha, Martins, and Machado (2018a) can, in part, be considered an exception as it provides a mechanism for extending to related concepts.

We propose that a visual conceptual blending should not only result in a visual blend produced for a given concept but instead be complemented by a much more developed conceptual layer (e.g. accompanied by other data such as a name or a description).

Despite providing valuable clues on the direction towards a possible model on visual conceptual blending, these systems cannot be considered as one. In our opinion, they fail to address several topics that we believe are important when building visual conceptual blends.

In this paper, we aim to take a step closer to outlining a model for visual conceptual blending that can be instantiated in a fully operational computational system. Nonetheless, our main goal is to provide a roadmap rather than a final blueprint, providing a broad description that mentions all the topics that we deem important to build such a model. The authors admit that this roadmap is most likely incomplete and will need to be improved in future iterations.

From Metaphors to Visual Blending

According to Peterson (2018), in a metaphor a source domain is recalled for comparison, and aspects of its identity are mapped onto a target domain.

When it comes to visual metaphors, they consist in the combination of objects that establish a similar comparison. However, their interpretation is harder as they are not direct in conveying messages (Petridis and Chilton, 2019), often lacking visual cues on what is the source and the target (Forceville, 1994). Upon observing a visual metaphor, one is able to recognise an object but at the same time notices something strange about it, causing a search for meaning (Chilton, Petridis, and Agrawala, 2019).

Despite not all visual blends being visual metaphors, most research is conducted in relation to advertisement and focuses on them (e.g. Phillips and McQuarrie, 2004; Gkiouzepas and Hogg, 2011).

Components and Types of Visual Blending

Chilton, Petridis, and Agrawala (2019) define a visual blend as having the following properties:

- Two concepts are given as input and each is mapped to an object that visual represents or symbolises it;
- The visual blend is an object that integrates the two initial objects, in a way that they are still recognisable and allow the user to infer an association between the concepts.

Regarding blend types, one categorisation was done by Phillips and McQuarrie (2004), who propose three types of blending of increasing complexity: juxtaposition (depict both source and target), fusion (the domains are combined) and replacement (one of the domains is omitted).

Peterson (2018) presented an expansion to this typology: identification – one domain pictorial, other textual; pairwise juxtaposition – both entities complete and separate, equal to juxtaposition by Phillips and McQuarrie (2004); categorical juxtaposition – source amidst target set, relates to that category concept; replacing juxtaposition – one entity breaks a set of selfsame entities, replacing one instance; replacement – one entity is absent and must be imagined by the viewer using contextual cues, equal to replacement by Phillips and
McQuarrie (2004); replacing fusion – part of one entity is replaced by another entity or part of it; and fusion – two entities are fused together to form a hybrid.

This typology seems easier to employ and to better match what is done by several authors working on visual blending. For example, Cunha, Martins, and Machado (2018a) use the term “replacement” but what their system performs – using an emoji to replace part of another – is more aligned with “replacing fusion” as defined by Peterson (2018).

Analysis to Visual Blends

According to Pollak et al. (2015), there are still many open questions regarding the production of blends. By investigating human creations and identifying patterns, it is possible to address these questions and possibly find a direction for the blending process, eventually allowing the automated generation of blends (Pollak et al., 2015). Joy, Sherry Jr, and Deschenes (2009) conducted an analysis of blends based on human perception by analysing conceptual blending in advertising. Bolognesi, van den Heerik, and van den Berg (2018) built a corpus of visual metaphors that have been analysed and annotated on different dimensions of meaning. Petrakis and Chilton (2019) focus on how people interpret visual metaphors of advertisements and identified causes for misinterpretation.

For our work, the most interesting example of blend analysis was conducted by Martins et al. (2015), who conducted an online-survey questionnaire in which participants were asked to evaluate criteria assumed to be related to the quality of blends. Martins et al. (2015) used visual blends between two animals (see Fig. 2) and tried to identify what humans perceive as a good blend. These blends used fusion and were focused on perceptual features, e.g. color, texture, or pattern.

Upon analysing the blends (see Fig. 2), one observes that colour cannot be considered the main reason for conducting the blend – i.e. animals are not blended on the basis of similar colour – but as a way to produce a good blend by achieving a fully integrated blend. Nonetheless, in some blends colour alignment of the input animals seems occur (e.g. pengwhale or guinea lion). In the same way proportion is also not the ground for blending, as several examples exist of strange proportion between head and body (e.g. snorse). It leads to the conclusion that the selection of the input animals was conducted without any apparent reason or conceptual grounding. Regarding the mapping that leads to the blend, one can see that it is mostly based in element category similarity (e.g. head of the snake is mapped to the head of the horse). Nonetheless, in (Martins et al., 2015) special attention is given to elaboration: name building and context creation.

Another example of blend analysis is described by Chilton, Petrakis, and Agrawala (2019), who stated that they observed blend examples and tested theories to come up with a design pattern – they identified shape as a particularly important feature in visual blending. Based on this, they developed a workflow for producing visual blendings based on an abstract structure: blend two objects that have the same basic shape but other identifying visual features. This example contrasts with the one from Martins et al. (2015) as they use a completely different feature. In addition, whereas Martins et al. (2015) only used blends of animals (fusion blend type), Chilton, Petrakis, and Agrawala (2019) analysed visual blendings of objects based on replacing fusion type. Similar studies are needed with other types of blend.

Roadmap for Visual Conceptual Blending

In this paper we present a model for the production of visual blendings with a strong conceptual grounding. In a process of visual conceptual blending, despite the output being a visual blend, it does not merely consist in the task of producing a merge of two initial visual representations. Instead, core of the process has to do with conceptual reasoning, which serves as base for the actual process of visual blending. This contrasts with the description given by Chilton, Petrakis, and Agrawala (2019) for the constituents of a visual blend, presented in the previous section.

In visual conceptual blending, the focus is not the transformational task of mixing two images but the whole process of producing visual blends that are based on a conceptual reasoning and present themselves as a result of a knowledge-based process. In fact, from our perspective, the output of a process of visual conceptual blending is not only an image but also a set of conceptual elaborations. A visual conceptual blending has context, it is grounded on a justification which should indicate the relevance of the blend. It can also be given a name that may not even be aligned with the original concept.

In this section, we outline a model for the production of visual conceptual blends. Our roadmap is composed of four main stages: (i) Conceptualisation; (ii) Visual Blending; (iii) Quality Assessment; and (iv) Elaboration.

Despite presenting it as a series of stages that may seem to occur in a linear sequence, the reader should understand that order may vary, not being fixed and allowing the repetition of some of the stages. These stages are somehow aligned with the three operations proposed by Fauconnier and Turner (2002): composition, completion, and elaboration.

Conceptualisation

By just focusing on the visual blending process, we can generate an infinite number of possibilities. However, these are not guaranteed to be grounded on knowledge. For example, for the visual blend between a snake and a horse, instead of the logical category-category mapping between heads (seen in Fig. 2) it would be possible to produce a blend in which head of the snake replaces the tail of the horse. However, such blend would have a very low conceptual grounding as no apparent logical mapping was performed. Conceptualisation is what distinguishes mere generation from something with a strong conceptual grounding (resultant from a process of reflection) and consequently visual blending from visual conceptual blending.

Conceptualisation can occur in at least two stages: selection of input concepts and mapping between previously given ones. Most examples described in this paper fall under the latter case – the input concepts to use in the blend are...
previously chosen. However, in a general model an initial step may be to identify potential candidates for a visual conceptual blending (Gonçalves et al., 2015). As such the topics addressed in this section can mostly be applied to these two situations.

In fact, the process of conceptualisation may lead to the retrieval of related concepts for the production of the visual blend (e.g. Veale, 2012; Cunha, Martins, and Machado, 2018a). In such case, there may not be a direct relation between the origin concept and the visual blend. This is specially evident when the origin concept consists in one word – it is necessary some sort of expansion to provide a foundation for a processing of visual blending to occur. In these cases, the origin concept can be visually represented by resorting to related concepts, e.g. freedom represented using a related concept “universal right” (see Fig. 1). In turn, the interpretation of the resultant visual blend can even lead to a third concept, for example “travel the world”. Therefore, the levels of conceptualisation of a visual blend can vary. In fact, the process of conceptualisation can reach high degrees of complexity – e.g. using a process of conceptual blending based on structural alignment techniques to produce analogies from structures such as mental spaces (see Fig. 4) – such was used by Cunha et al. (2017).

On the other hand, a process of visual conceptual blending can have different motivations and therefore different goals. For example, the process can be used for concept representation, in which case a literal representation may be preferred. Another possibility may be the production of visual metaphors, in which case the goal will be more creative.

In the end, the conceptualisation stage consists in answering the question: what is behind the blend? How this question is approached depends on the starting point. For example, if we already have a two word concept it is more related to how we blend the two concepts – finding a justification for a blend. If the starting point is a one word concept, we face a somehow open search for potential blends – which is good if we have enough knowledge.

Several characteristics can motivate the process of blend, e.g. conceptual features (e.g. name or affordances) or perceptual features (e.g. shape or colour).

**Grounding the blend: perceptual features** One way of grounding the blend is by using perceptual features, e.g. shape or colour. The usefulness in perceptual features is especially relevant when these include prototypical elements (Johnson, 1985) – i.e. what most identifies a given concept (e.g. the nose and the tails in a pig). An example is the work by Karimi et al. (2018) in which blend possibilities are found using a process of conceptual shift based on shape comparison.

Obviously, these characteristics are very dependable on the representation used – e.g. if only black and white images are to be used, colour loses relevance. The mappings based on perceptual features always depend on the situation.

**Grounding the blend: affordances** Another way of finding blend possibilities is related to affordances and their modelling using, for example, image schemas. Such may help in guiding how the visual blending should be conducted – e.g. using the schema CONTAINMENT with icons of “money” and “building” to represent bank (Cunha, Martins, and Machado, 2018b; Falomir and Plaza, 2019).

**Grounding the blend: naming** A third possibility has to do with the name – e.g. finding homophones such as “waste of money” and “waist of money”. As an example, Veale and Al-Najjar (2016) explore the invention of colour names.

**Visual Blending**

Existing visual blending systems can be divided into two groups based on the type of rendering (see Fig. 1): photorealistic, e.g. (Xiao and Linkola, 2015), and non-photorealistic, e.g. (Cunha et al., 2017). These two types have great differences in terms of how the visual blending process occurs. A photorealistic visual blending may require computer vision and image processing techniques, whereas a non-photorealistic visual blending that uses fully scalable vector graphic is much easier to conduct (Cunha et al., 2017).

In either case, a process of visual blending involves two main decisions: which objects to combine and how to combine them.

**Connection between Conceptual and Visual** Most of the visual blending examples that are grounded on a process of conceptual blending consist in a simple visualisation of the blend, e.g. (Pereira and Cardoso, 2002). An exception can be seen in (Cunha et al., 2017), in which two types of net-

![Figure 3: On the left is a the representation drawn with the elements identified; On the right is the result of the conversion into fully scalable vector graphic (Cunha et al., 2017)](image-url)
work structures were used: one corresponding to the mental spaces of the input concepts and another corresponding to the visual structure of the visual representation (see Figs. 3 and 4). The two types of structure were aligned to produce visual conceptual blends. In this case, the system was considered a hybrid blender, as the blending process starts at the conceptual level and ends at the visual one. However, this situation is uncommon, as in most cases it is not possible to align the conceptual layer with the visual one – such data would have to be manually built. One possibility would rely on an analysis of the images to produce a network structure (structure extraction). This is more or less easy to implement in fully scalable vector graphics but on raster images it would have to use techniques such as concept detection (Zhou, Jagadeesh, and Piramuthu, 2015).

In any case, in the same way the visual level is based on what is produced on the conceptual level, the conceptual level also needs to take into account the character and features of the representations being used.

Questions of semiotics In addition to the simple exchange of parts, there are several aspects that have to be taken into consideration. Cunha et al. (2015) address this issue and provide some guidance on how color, shape and other visual aspects may affect meaning. Moreover, one should also consider elements such as modifiers (Cohn, 2007) and bear in mind that there are cultural differences that ultimately have an impact how a visual blend is interpreted (Cunha et al., 2015).

Type of Blend Each type of blend is suitable to different types of concepts and visual representations. As such, the choice of which blend type to use should take certain aspects into consideration. First, it should consider the relationship between the categories of the concepts being blended. For example, it is completely different blending “dinosaur” with “park” and “dinosaur” “fish”. The former case involves an animal and a location, which makes it more suitable to have a juxtaposition. In the latter case, both concepts are animals and, as such, a fusion might be more appropriate.

Then, since the process of blending involves visual representations (e.g. icons), the appropriateness of blend type also varies depending on type of representation being used. For example, in the “dinosaur” “fish” the animals are very different and that will have an impact on how the blend is conducted. Moreover, it is completely different to blend two representations that show the full body of the animal and one that shows a full body and another that only shows the head. For the blends shown in Fig. 2, the author mostly likely had to carefully select the images that better matched one another.

Quality Assessment While producing blend, it is important to have a measure of quality. If we have a system that produces several individuals, a measure of quality is crucial to identify good solutions. In certain situations the blend production can be considered an open-ended problem, in which case including the user in the cycle may provide some advantages. Nonetheless, several types of quality assessment exist – some may be more suitable for certain goals than others.

In fact, Martins et al. (2015) poses several questions regarding quality assessment: “How ‘semantically far’ should the input spaces be to produce a good blend?”, “Is there a correlation between the quality of blends and the number of elements for projection?” or even “Are all the optimality principles required to produce good blends?”. In this section, we present some types of quality measures that can be used to assess how good a blend might be.

Argumentation Confalonieri et al. (2015) proposed the use of argumentation to evaluate and iteratively refine the quality of blended computer icons. The authors introduced a semiotic system, which was based on the idea that signs can be combined to convey multiple intended meanings. Despite this, no evidence of a possible implementation was provided.

Optimality Principles Fauconnier and Turner (1998) proposed a list of optimality principles that can guide the process of conceptual blending. These principles are not trivial to computationally model and are normally used at the conceptual level. Nonetheless, it is also possible to use them to validate the blend on the visual level, as Kowalewski (2008) demonstrated by analysing the formation of logos and product names in terms of usage of optimality principles. Even though these principles are considered as responsible for generating consistent blends (Martins et al., 2015), they should not be regarded as “rigid laws” but as flexible guidelines (Kowalewski, 2008). We provide a description of these principles below:

- Integration: the blend must constitute a tightly integrated scene that can be manipulated as a unit. It should be a coherent, self-contained and unified structure, and not a loosely knit combination of random elements (recognized as a whole). Integration is identified by Martins et al. (2015) as the most important principle;
- Topology: the elements projected into the blend should maintain the same neighbourhood relations as in the input space. Even though Martins et al. (2015) indicate that topology as not relevant, according to Kowalewski (2008) it can be useful for example in terms of spacial organisation by placing elements in the blend according to the configuration of one of the input visual representations (e.g. maintaining the existence of a central element, laying new elements according to center-periphery scheme);
- Web: the blend as a unit must maintain the web of appropriate connections to the input spaces, so that an event in one of the input spaces implies a corresponding event in the blend;
- Unpacking: this principle takes the perspective of the reader and consists in the easiness of reconstructing the inputs and the network of connections from the blend. The input concepts should be recognisable from the elements of the blend as the input visual representations or parts of them. One example of this can be seen in the use of prototypical parts in the blend;
- Relevance (or Good Reason): if an element appears in the blend it should have some kind of significance/meaning.
Two other principles are “Intensifying Vital Relations” and “Maximising Vital Relations”. However, in this context we could not provide a clear usefulness for them. In addition to being sometimes vague and difficult to implement, not all the principles are compatible with each other (Martins et al., 2015). Moreover, choosing some over others may lead to more or less creative blends (Martins et al., 2016).

**Visual Analysis** Assessing quality can also concern visual aspects. Two examples are: overall complexity and area exchanged (Cunha et al., 2020). It is important to mention that some aspects are easier to apply in a visual blending with layered images. For raster images other aspects may be more appropriate.

**User Perception** Despite the importance of all the topics already mentioned, the quality of visual blend will always depend on user perception and interpretation, and may perform poorly even if the blend is conceptually grounded. Providing a way for the user to interact with the system would make it so that improvements could be made to increase the likelihood of a correct interpretation. A method used by some systems (e.g. Cunha et al., 2019) is Interactive Evolutionary Computation, which consists in including the user in the task of fitness assignment and evolving solutions that match their preference.

**Elaboration** A big part of the conceptual process may occur after the visual blending is done – consisting of an elaboration. This elaboration and consequent interpretation may in turn serve to provide justification for the previously done visual blend and also as a way to improve it – resulting in a return to a previous stage for a new iteration.

**Naming** One example of elaboration is the production of names. Pollak et al. (2015) presented a prototype for name generation based on an investigation focused on the principles of creating lexical blends based on visual blends (blended animals). Pollak et al. (2015) identified the following mechanisms used in name formation: L1-concatenation blends; L2-portmanteaux (e.g. rabbear for rabbit and bear); L3-blending based on visible characteristics; L4-blending using background knowledge and L5-bisociative blends (e.g. mickey the bear for mouse and bear). These techniques can be used for other blends that do not use animals.

**Descriptions** In addition to names, there is also the potential to produce descriptions based on the visual blend. Techniques such as image captioning (Feng et al., 2019) may be used for this purpose. Ideally, a system that produces descriptions could produce an elaboration on the context of the blend. For example, mixing two animals leads to questioning the context of the hybrid animal: Where does it live? What does it eat? How does it behave in relation to other animals? All these questions would need to be addressed using a process of conceptual blending by getting characteristics from the two mental spaces. An example can be observed in the concept clown fish: does it live in the sea and looks like a clown? does it live in a circus and looks like a fish? Obviously, one of the situations has a higher likelihood, which makes it more plausible; but the surprising nature of the other option makes it so that in terms of creativity it has much more potential.

Moreover, a creative system like the one we are proposing would have great advantages in providing the user with explanations for the produced blends. The descriptions can be seen as such and used to make the process of blending clearer to the user (Cook et al., 2019).

**Towards Implementation** In this section, we briefly analyse existing approaches using the topics presented in the previous section. Then, we provide some guidelines that we believe should be taken into account when building a general model for visual conceptual blending.

**Analysis to existing blend systems** In order to summarise existing approaches, we conducted an analysis to several systems that address visual blending (see Fig. 5). The analysis was made in terms of type of system (creativity-support tools vs creative system), type of blend (replacement, juxtaposition or fusion) and guide of blend (shape or conceptual). The analysed approaches were: Steinbrück (2013); Xiao and Linkola (2015); Ha and Eck (2017); Cunha et al. (2017); Karimi et al. (2018); Cunha, Martins, and Machado (2018a); Chilton, Petridis, and Agrawala (2019); Zhao et al. (2020). None of them addresses all the topics we mention on the model.

**General Model** Having analysed existing systems, we now present a set of aspects that, in our opinion, will be key in implementing a general model for visual conceptual blending.

**Modularity** Most of the systems described before work in an individual way with no connection to others. An exception is Vismantic (Xiao and Linkola, 2015), which is integrated in a platform for workflow management – ConCreTeFlows. Martins et al. (2019) focus on this platform...

![Figure 5: Analysis to existing visual blending approaches](image-url)
and present an example of how it can be used to develop CC software components that can shared, used and reused to produce complex computational pipelines. We believe that an implementation of a general model for visual conceptual blend will profit from using such modular approach, allowing multiple users to contribute to the system.

**Multi-approach** In addition to having several modules that deal with different tasks, as we have seen earlier, there are several methods that can be employed for each of the tasks (e.g. conceptualisation can be based in perceptual features, affordances, etc.). The suitability of these methods often depends on the type of problem at hands (i.e. the characteristics of the blend) and, as such, no optimal approach exists. A solution to this multi-approach situation is to follow a similar strategy to the one presented by Cardoso et al. (2015) – using a global workspace and a number of components that compete for access to it. Each component could be seen as an agent. At each time, the agent that is able to produce the most relevant output is given access to the workspace. This would consist in having solutions being produced by each of the agents and finding the best.

**User centred** The quality of a visual blending always depends on user perception, thus being of open-ended nature. As such, the user should be viewed as having a central role. The modular approach suggested earlier is obviously dependent on having a user interacting with the platform to build the a pipeline of components. We go one step further and propose that the user should also have an active role in producing the visual blends.

First, the interaction with the user has great potential to be explored as it can be used to iteratively improve the quality of the blend, both visually and conceptually. This would consequently have an effect on which approach is used at each task, depending on the user evaluation. Moreover, the user would guide the blend production in terms of improving second-order features (e.g. color) or even extending the conceptual reach when no blends can be produced with the existing knowledge.

Another possibility is to provide the user with a way of selecting the creativity degree they want for the blend – low creativity resulting in literal representations and high creativity in more metaphorical results.

**Conclusion and Future Work**

In this paper, we focused on visual conceptual blending. We started by providing the reader with a general view on current research on visual blending. Then we presented a proposal of a model for the production of visual conceptual blends. This model can be instantiated into a modular system, in which the different stages of blend production occur in an iterative manner, allowing the user to go back to improve the blend and its elaboration. Future developments concern the implementation of the proposed model, as well as the establishment of collaborations with researchers who develop work in areas related to the identified topics.

**Acknowledgements** This work is partially funded by the FCT - Foundation for Science and Technology, I.P., within the scope of the project CISUC - UID/CEC/00326/2020 and by European Social Fund, through the Regional Operational Program Centro 2020, and under the grant SFRH/BD/120905/2016.

**References**


Cunha, J. M.; Gonçalves, J.; Martins, P.; Machado, P.; and Cardoso, A. 2017. A pig, an angel and a cactus walk into a blender: A descriptive approach to visual blending. In *Proc. of the Eighth Int. Conf. on Computational Creativity*.


Lemotif: An Affective Visual Journal Using Deep Neural Networks

X. Alice Li  
Georgia Tech  
xali@gatech.edu  

Devi Parikh  
Facebook AI Research & Georgia Tech  
parikh@gatech.edu

Abstract

We present Lemotif, an integrated natural language processing and image generation system that uses machine learning to (1) parse a text-based input journal entry describing the user’s day for salient themes and emotions and (2) visualize the detected themes and emotions in creative and appealing image motifs. Synthesizing approaches from artificial intelligence and psychology, Lemotif acts as an affective visual journal, encouraging users to regularly write and reflect on their daily experiences through visual reinforcement. By making patterns in emotions and their sources more apparent, Lemotif aims to help users better understand their emotional lives, identify opportunities for action, and track the effectiveness of behavioral changes over time. We verify via human studies that prospective users prefer motifs generated by Lemotif over corresponding baselines, find the motifs representative of their journal entries, and think they would be more likely to journal regularly using a Lemotif-based app.

Introduction

Our emotional well being is important. In part due to its subjective nature, it is difficult to find patterns in what we feel, how often we feel it, and what the source of those feelings tends to be. Without this assessment, it is difficult to tweak our choices to optimize our emotional well being. Meanwhile, innovations in artificial intelligence have produced powerful neural networks capable of sophisticated analytic and generative tasks. There exists great potential for machine learning to address human-centered needs, using creative interdisciplinary approaches to model subjective qualities like emotion which can be difficult to quantify.

In this paper we introduce Lemotif, an integrated natural language processing (NLP) and image generation system serving as an affective visual journal. Given a text-based journal entry describing aspects of the user’s day, a multi-label classifier building upon the Bidirectional Encoder Representations from Transformers (BERT) language model (Devlin et al. 2018) extracts salient topics and associated emotions from the provided input. An image generation algorithm then creates motifs conditioned upon the detected topics and emotions; we offer several representation styles, including a neural network trained on abstract art. We evaluate Lemotif qualitatively via human studies assessing whether (1) the topic-shape and feeling-color mappings are meaningful to users, (2) subjects favor the generated motifs over corresponding baselines, (3) subjects consider the generated motifs representative of their journal entries, and (4) subjects would engage positively with a Lemotif-based app, such as feeling like the app would encourage them to journal more regularly. The NLP model is evaluated as a multi-label classifier calculating F1 and nor-
malized accuracy through cross-validation. We report favor-able results on all fronts. Our code and trained models are available at https://github.com/xaliceli/lemotif. A demo is available at http://lemotif.cloudcv.org.

Related Work
Our work processes text to extract key topic and emotion labels, maps these abstract concepts to visual entities (shape and color), and generates a visual depiction of the input text according to the extracted labels. In this section we discuss prior work relating to each individual component as well as our overall goal of creatively summarizing journal entries.

Journaling Tools  Our work is motivated by psychological research indicating that writing about emotions can support mental health (Pennebaker 1997). Most existing journaling tools allow users to log their lives without focusing on identifying themes or patterns, while we aim to make associations between a user’s feelings and aspects of their life apparent. When journaling apps claim to be "visu- al" (e.g., HeyDay), they typically refer to allowing visual input modalities such as images and videos. Our work produces a visual modality as an output. Life Calendar (journallife.me) comes closest to our approach, showing a single-colored dot (red, yellow, or green) for each week that captures the mood of the user in that week (negative, neutral, positive). This allows one to find correlations between time of month or year and emotion (e.g., happier in the summer). But it does not help identify sources of nuanced emotions on a day-to-day basis. In our experiments, we compare our motifs to a visualization that mimics this and find that subjects strongly prefer our nuanced and creative visualizations.

Natural Language Processing Our task of identifying topics and emotions from text is related to existing work on keyword extraction and sentiment analysis, though sentiment analysis is commonly approached as a binary or tri-nary (e.g. "positive", "neutral", "negative") problem. Recently, BERT (Devlin et al. 2018) and GPT-2 (Radford et al. 2019) have successfully pre-trained NLP models on large unlabeled text datasets, learning language representations that can then be generalized to downstream tasks. We fine-tune BERT (pre-trained on the BooksCorpus and En-glish Wikipedia datasets) on a custom dataset for our specific task of identifying up to 11 topics and up to 18 associated emotions, a form of aspect-based sentiment analysis (Liu 2012). To the best of our knowledge, no off-the-shelf system currently exists with a topic and emotion set as broad and granular as ours.

Visual Representation Our work draws upon existing research on common associations between colors and emotions (Nijdam 2005). Studies have also indicated associations between the visual qualities of lines and the emotions they evoke (Poffenberger and Barrows 1924). There exists fascinating work on projecting a spectrum of human emotions on an interactive map with associated video clips that elicit those emotions (Coven and Keltner 2017) and to audio gasps that people make when expressing those emotions (Coven et al. 2018). (Salevati and DiPaola 2015) studied the emotions evoked in viewers of paintings generated by a creative AI system, while (Alvarez-Melis and Amores 2017) collected emotional labels for artworks and trained a generative adversarial network to synthesize new images conditioned upon emotional labels. We extend the foundational idea that emotions can be represented in visual form by using a relatively rich set of 18 nuanced emotions as a core design principle of Lemotif. The use of recognizable icons to represent topics is also a common feature in popular note-taking apps such as Notion (notion.so); we take a simi-lar approach, using shapes to represent topics in Lemotif.

Image Synthesis Approaches in multi-modal AI generating natural images from their language descriptions (Reed et al. 2016) using generative adversarial networks (GANs) (Goodfellow et al. 2014) are relevant. Convolutional neural networks comprising an encoder block for feature extraction and a decoder block for synthesis and reconstruction, or an "autoencoder" model (Hinton and Salakhutdinov 2006), have been widely used for image-to-image translation tasks, including image super-resolution taking small samples as inputs and reconstructing larger outputs with greater detail (Dong et al. 2015). We take this super-resolution approach for our autoencoder visualization style, with more details in the following section.

Computational Creativity Our system combines human inputs (text and visualization parameters) with stochastic computational processes to generate appealing images. Our visualizations draw from generative art (Galanter 2003), using algorithms for aesthetic purposes. Our approach also relates to prior work indicating the effectiveness of creative computational systems for therapeutic use and emotional well-being (Cheatley, Moncur, and Pease 2019).

Approach
Below we describe our approach to processing journal entries, mapping concepts to visual content, and generating representative motifs.

Natural Language Processing Our NLP objective is to take free-form text input and predict salient topic and emotion labels. To that end, we fine-tune BERT to serve as a multi-label classifier. We use the BERT-Base model containing 12 encoder layers, 768 hidden units per layer, and 12 attention heads for a total of 110M parameters (Devlin et al. 2018). To BERT-Base we append a fully-connected multi-layer perceptron containing 768 hidden units with sigmoid activation and 29 output nodes corresponding to our 11 topics and 18 emotions. We fine-tune this model on our dataset of text samples with user-annotated labels (more details in the Dataset section), optimizing over sigmoid cross-entropy loss. Labels above a set probability threshold are returned as the salient topics and associated emotions; we use 0.2 as our threshold, chosen through cross-validation. These labels are then used as inputs for the image generation algorithms, such that each motif represents one topic with the highest probability and a set of up to four emotions with the highest probabilities.
afraid    angry    anxious    ashamed    awkward    bored    calm    confused    disgusted    excited    frustrated    happy    jealous    nostalgic    proud    sad    satisfied    surprised

Figure 3: Colors used to represent various feelings or emotions.

Topics and Emotions

Human experiences are complex, multimodal, and subjective. A system that identifies and visualizes abstract content about an individual’s emotional life must be both comprehensive and intelligible, addressing most common themes in life through discrete labels while representing this information in a format humans recognize and approve of. Below we outline our approach to identifying our target labels and mapping these concepts to visual representations.

Topics The 11 topics in our pre-defined list are shown in Fig. 2. This list was determined by a mix of brainstorming and searching online for what topics users typically talk about in their journals. As part of our evaluation, we asked survey participants in an Amazon Mechanical Turk (AMT) study if they felt a topic they would like to talk about was missing. 99 subjects out of 100 said this list was sufficient. One user suggested adding pets as a topic.

Emotions The 18 emotions in our pre-defined list are shown in Fig. 3. This list was curated from (Cowen and Keltner 2017) and our assessment of what emotions are likely on a day-to-day basis. Again, as part of our evaluation, we asked users from the same AMT study described above if they felt an emotion they would like to talk about was missing. All 100 subjects said the list was sufficient.

Shapes for topics Lemotif uses a pre-defined mapping from topics to visual icons of shapes depicting that topic. These are shown in Fig. 2. To identify our list of icons, we searched The Noun Project (http://thenounproject.com, a resource containing millions of binary icons created by designers) for icons relevant to each of the topics within our label set. From the relevant icons, we selected those that are not visually complex so the generated motif is clear. We binarize, crop, and resize each image to a canonical size, post-processing icons to retain only the outer shape. The resulting icons are shown in the bottom row of Fig. 2.

Colors for emotions Lemotif uses a pre-defined mapping from emotions to corresponding colors associated with that emotion, as shown in Fig. 3. These colors were selected based on common associations (e.g., dark red for angry) as indicated by studies (Nijdam 2005) while making sure that each color is visually distinct (Trubetskoy 2017).

Image Synthesis

Taking a set of labels extracted by the NLP model consisting of topics and emotions, Lemotif generates image motifs depicting these salient themes in visual form. Acknowledging that creative preferences are inherently subjective and individual, we offer six creative visualization styles described next. The generated visualization image is then bounded by a shape icon representing the relevant topic. The human user exercises creative input in selecting a motif style and adjusting various input parameters according to personal taste, while the algorithm produces unique motifs with stochastic variations for each generated image.

Autoencoder We train a convolutional neural network designed as an autoencoder, taking a low-resolution image as its input and predicting a high-resolution version of the same image as its output (generated output shown in A6 in Fig. 4). We design our model to perform this form of super-resolution because we want to provide the model a set of colors representing emotions and allow the model to generate creative and stochastic detail — in other words, we want the model to begin with limited information (colors) and learn higher-resolution artistic representations of the provided colors. Our model consists of three residual blocks (He et al. 2016) encoding a 16x16 input image to feature space and a standard convolutional decoder architecture containing 2D Convolution + BatchNormalization + LeakyReLU blocks producing a 256x256 output image.

For our research study, this model is trained on a dataset of 14,621 abstract paintings from WikiArt (downloaded from https://github.com/cs-chan/ArtGAN), randomly cropped to 256x256 high-resolution ground truths and resized to 16x16 low-resolution inputs. In training, we minimize mean squared error loss between the generated output and the original cropped image. In inference, we randomly populate a 16x16 image with pixel colors corresponding to the provided emotions, producing an output image in the style of abstract art in our target colors. This model can also be trained on different datasets containing artworks from varying artists or artistic movements to produce motifs of diverse styles.

Carpet Carpet (A3 in Fig. 4) divides the image into a grid, repeatedly placing parallel lines in each cell of the grid at one of four possible angles and filling in the resulting connected regions with a random color from the set of colors.
Figure 4: Our six proposed visualizations (A1...A6) and seven comparison baselines (B1...B7).

associated with the detected emotions. Users can adjust the thickness and angles of the lines placed and the grid size that the canvas is divided into.

**Circle packing**  In circle packing (A1 in Fig. 4), we fill a blank region of a given shape with circles of differing sizes, each filled with a random color out of the colors associated with the detected emotions. We start with a set of circle radii and the desired number of circles to be placed in the region for each radii, which can be adjusted by users to taste. Starting from the largest size, we sample a random location in the shape region. If a circle can be placed there without any part of the circle falling outside the region or overlapping an existing circle, we draw the circle. If not, we sample another random location and try again until a max number of trials is reached or the circle is successfully placed. This is repeated for the specified number of circles to be placed for each size.

**Glass**  Glass (A5 in Fig. 4) attempts to mimic the appearance of stained glass by placing an assortment of icons in the topic shape at differing colors and opacities. By overlapping the canvas region with translucent icons across multiple passes, a random pattern of colors and shapes emerges. Users can customize the number of passes, how densely or sparsely icons are placed, and the distribution of icon sizes.

**Tile**  Tile (A4 in Fig. 4) divides the image into a grid, randomly placing a line in each cell along one of two diagonals and filling in the resulting connected regions with randomly chosen colors corresponding to the detected emotions. Users can adjust the grid size, line width, and probability that each one of the two diagonals is picked.

**String Doll**  String Doll (A2 in Fig. 4) draws quadratic bezier curves that connect two random points on a blank top-}

ical shape’s boundary, without the stroke going outside the boundary of the shape. As the control point of the quadratic bezier curve, we take the mid point of the two end points and add zero-mean gaussian noise to it. The standard deviation of the gaussian is set to 20% of the size of the canvas. The strokes are colored uniformly randomly by one of the colors corresponding to the emotions detected in the user’s journal entry for that topic. The width of the stroke is sampled from a distribution of sizes. To add some texture to the visualization, each stroke is overlaid by a stroke that is lighter or darker in color and a quarter of the original stroke’s width. Users can adjust the number and width of strokes and the standard deviation of the gaussian controlling the placement of each quadratic bezier curve’s control point.

**Dataset**

We collected 500 journal entries from 500 anonymous subjects on Amazon Mechanical Turk (AMT), responding to the prompt “What were salient aspects of your day yesterday? How did you feel about them?” Figure 1 contains an example of one entry from a respondent. Each journal entry contains up to three text samples describing different aspects of the subject’s day; referred to as sub-entries henceforth in this paper. We asked subjects to annotate each sub-entry by selecting its associated topics and emotions from a drop down list populated with our set of topics and emotions, serving as ground truth labels for our NLP model evaluation. Our dataset is available at https://github.com/xaliceli/lemotif.

For entries in our dataset where subjects wrote meaningful responses relevant to the prompt, the mean entry (containing up to three sub-entries) was 507.6 characters (100.6 words) long; on average, each entry included 5.9 emotions.
Figure 5: Distribution of topics and feelings in our dataset.

Figure 6: Cross-validation F1 and normalized accuracy statistics by varying probability thresholds used to indicate a positive prediction. Line at 0.2 represents the threshold we select for inference.

Experiments and results

Evaluating icon and color choices We showed subjects on AMT our list of 11 topics and a randomly ordered list of the 11 icons shown in Fig. 2. Subjects were asked to assign each icon to exactly one topic. 170 subjects performed this task. Given a topic, the right icon was picked 69% of the times (mean across subjects), compared to the random chance probability of ~9%. If we assign a topic to the icon that was picked most often for that topic (majority vote across subjects), the accuracy is 82%. For a given topic, we sort all icons by how often they were selected across subjects. We find that the right icon falls at rank 1.27 out of 11 (on average across topics). The right icon falls in the top 20% of the sorted list 91% of the time across topics, and in the top third of the list 100% of the time. Overall, subjects appear to find our topic-icon mapping intuitive and natural.

We ran a similar study to evaluate our feeling-color mapping shown in Fig. 3. This is a more challenging task because (1) icons have descriptive shapes that can be recognized as objects with semantic meaning, while colors are significantly more ambiguous, and (2) there are 18 feelings and colors as opposed to fewer topics and icons. Note that the choice of colors (and icon) being intuitive and natural to users is a bonus, but not a requirement; as seen in Fig. 1, the topics and feelings are explicitly listed on the motif. 99 subjects participated in this study; because this task was more involved, fewer AMT users elected to participate compared to the topic-icon evaluation. We find that given a feeling, the right color was picked 15% of the time (mean across subjects). Chance performance would be ~6%. If we assign a feeling to the color that was picked most often for that feeling (majority vote across subjects), the accuracy is 33%. For a given feeling, we sort all colors by how often they were selected across subjects. We find that the right color falls at rank 5.28 out of 18 (on average across feelings). The right color falls in the top 20% of the sorted list 61% of the time across feelings, and in the top third of the list 67% of the time. Overall, this shows that despite the mapping being ambiguous and subjective, subjects do find an intuitive and natural signal in our feelings-color mappings as well.

Evaluating natural language model We trained our NLP model on our text dataset with user-supplied ground-truth labels. We performed cross-validation across five train-test splits (80% train, 20% test) to calculate normalized accuracy and F1 metrics comparing ground truth versus predicted labels across the full dataset of 1,473 text samples (sub-entries). Normalized accuracy is the mean between true positive and true negative rates. F1 (also known as F-score) is the harmonic mean of precision and recall. Recall that our NLP model outputs multi-label probabilities between 0 and 1. Figure 6 shows normalized accuracy and F1 scores for various probability thresholds above which a label is counted as a positive classification. At our chosen threshold of 0.2, our model has a normalized accuracy of 82% and an F1 score of 0.62, compared to random chance values of 50% and 0.5. Since different thresholds yield similar accuracies, we use 0.2 during inference partially based on experimentation using new and arbitrary input samples. Because our dataset is fairly small with an uneven distribution of positive labels across topics and emotions (see Fig. 5), our model may not perform well on new samples referring to topics or emotions underrepresented in our dataset. We expect a larger training set would improve model performance and allow us to choose positive classification thresholds more empirically based on each label’s expected frequency.

Evaluating creative motifs The generated motifs should (1) separate the topical sources of emotions, (2) depict these sources visually, (3) depict the emotions visually, and (4) be creative and attractive. We design several baselines that allow us to measure the importance of each factor by removing one factor at a time. To keep the number of comparisons manageable, we create baselines only for circle packing (Fig. 4 A1) and string doll (Fig. 4 A2).
We start with our standard motif and remove the shape depiction, retaining the creative design, color depictions, and separate topic depictions. Baselines B1 and B2 in Fig. 4 show motifs as squares, without a topical shape.

We can also start with our standard motif and remove the creative design, maintaining the shape and color depictions and the topic breakdown. B3 in Fig. 4 shows motifs with solid colors instead of creative designs.

B4 in Fig. 4 removes both shapes and creative designs, showing squares filled with solid colors.

B5 in Fig. 4 starts with B3 (no topical shapes or creative designs) and further removes emotion-specific colors. Instead of using a different color for each of the 18 feelings, we use just red, yellow, and green to depict negative, neutral or positive feelings. We mapped afraid, angry, anxious, ashamed, disgusted, frustrated, jealous and sad to negative; awkward, bored, calm, confused, nostalgic and surprised to neutral; and excited, happy, proud and satisfied to positive. We use the majority label across reported feelings to pick a color for that topic.

B6 in Fig. 4 further removes topical shapes from B5, leaving squares colored in red, yellow, or green.

B7 in Fig. 4 finally removes the breakdown of individual topics from B6. Each day is depicted as one single red, yellow, or green square. As mentioned in the related work section, this mimics an existing app (Life Calendar) that shows a single colored dot for every week in the year.

To start, we combine these seven baselines with two of our proposed visualizations (Fig. 4 A1 and A2), giving us nine approaches to compare for this evaluation; later on we will compare all six visualization styles described in the Approaches section against each other and a smaller set of three baselines. We generate these visualizations for a subset of 100 journal entries from our dataset, using user-supplied ground-truth topic-emotion labels; each entry contains three sub-entries and their corresponding motifs. For parameters users can vary, such as line width and spacing, we set their values based on what we found most visually appealing and representative of each style’s overall design. We conduct pairwise evaluations on AMT. We show subjects a journal entry from our dataset, and all \( \binom{7}{2} = 36 \) pairs of visualizations. For each pair, we ask subjects “If you were using a journaling tool or app that automatically produced a visual summary of your day, which one of these visualizations would you prefer?” 936 unique subjects participated in this study, each providing us a rating for the 36 pairs for a single journal entry. Each journal entry was evaluated by 6 to 17 subjects, with an average of 9.4 and mode of 10.

By comparing pairs of the proposed approaches, we can evaluate the role of the four visual factors listed above. How often subjects pick B6 over B7 reveals how important it is for the motif to have a breakdown across topics. Similarly, comparing B3 to B6, B3 to B4, A1 to B1, and A2 to B2, indicates the importance of a topic being depicted by a shape as opposed to a generic square. Comparing B3 to B5, and B4 to B6, indicates the importance of each feeling being depicted by an emotion-specific rather than general color. We find that subjects prefer circle packing (A1) to string doll (A2) 72% of the time and therefore focus our evaluation of the creative aspect on A1. Comparing A1 to B3 and B1 to B4 reveals how much subjects prefer creative designs.

In Fig. 7, for each of the four factors, we show how often a visualization with that factor is preferred over a corresponding visualization without that factor (as described above). We show these statistics separately for subjects who were consistent in their preferences vs. those that had some contradictory preferences. We define consistent respondents as those whose preferences held across all pairwise comparisons, i.e. if \( a > b \) and \( b > c \), then \( a > c \). Presumably, subjects who provide consistent preferences are likely to be doing the task more carefully and/or have more clear preferences. We find that 36% of our subjects were perfectly consistent across the 36 pairwise comparisons. Across the board in Fig. 7, the four factors are preferred, especially for subjects who were consistent in their responses.

**Evaluating additional visualization styles** Having established that our four creative factors are generally preferred by human subjects, we next evaluate all six visualization styles (A1-A6 in Fig. 4) against a smaller set of three baselines: B3 (topical shapes and emotional colors with no additional creative style), B5 (topical shapes and positive-neutral-negative colors), and B6 (squares and positive-neutral-negative colors). Similar to the first evaluation performed, we generate 36 visualization pairs and ask respondents to select the style they prefer. 854 unique subjects participated in this study.

Fig. 8 shows user preferences across all six visualization styles and three baseline comparisons. Overall, the most preferred styles were the creative visualizations, consistent with what we saw in the prior evaluation. The one baseline comparison that performed comparably to random chance was B3, which includes both topical shapes and our full set of 18 emotional colors; though this baseline was not intentionally designed as a creative style, one could argue that...
(a) Percent of pairs in which style was preferred. P-value is from a one-sample t-test compared to null hypothesis of 50%. N reflects the number of pairs in which each style was compared.

Figure 8: Preferences across visualizations. Creative styles are shown in blue and baseline comparisons are shown in gray.

placing colors in equally distributed regions is a creative visualization. After all, there are entire artistic movements such as color field painting with similarly "flat" aesthetics. We also note that, when evaluating which style was each respondent's favorite (defined as the style that was most frequently preferred for each respondent), preferences are widely distributed across styles. For example, even though the autoencoder was preferred fewer than 50% of the time overall, 12% of respondents preferred it above all other styles, comparable to the 12% of respondents who most favored the glass visualization which scored highest in pairwise comparisons. The diversity of preferences highlights the personal nature of aesthetics and how the act of choosing a motif to use can be a creative decision in and of itself.

Evaluating engagement The real evaluation of a system like Lemotif is how users would engage with it — would users journal more regularly, feel more creative, and/or gain actionable insights from their motifs? Such a longitudinal evaluation is outside the scope of this paper. As a proxy, we ran two surveys on AMT. The first survey (study S1 with 100 unique and valid responses) described the concept of Lemotif to subjects and showed example circle packing motifs for reference. The second survey (study S2 with 99 unique and valid responses) directed subjects to a web demo asking them to write three pieces of text about their day and generated motifs in all six visualization styles based on labels automatically detected from their entries. Note that S2 evaluates our entire system end-to-end on free-form entries. Between S1 and S2, responses to the metrics shown are comparable, with the exception of "more enjoyable" and "get creative juices flowing" receiving lower scores in the end-to-end demo. Since S1 only describes Lemotif to users, they are free to imagine an ideal user interface. Moreover, assessment of the end-to-end demo would also suffer from errors in the NLP model, which is not perfect. We posit that a full app with attention to user experience design and our full set of customization options would likely score higher than S2 currently.

Table 1: Survey responses to Lemotif app

<table>
<thead>
<tr>
<th>Question</th>
<th>% Yes (S1)</th>
<th>% Yes (S2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Representative of entry?</td>
<td>NA</td>
<td>71%</td>
</tr>
<tr>
<td>Would use?</td>
<td>59%</td>
<td>56%</td>
</tr>
<tr>
<td>Make more enjoyable?</td>
<td>68%</td>
<td>59%</td>
</tr>
<tr>
<td>Would write more regularly?</td>
<td>61%</td>
<td>61%</td>
</tr>
<tr>
<td>Get creative juices flowing?</td>
<td>59%</td>
<td>51%</td>
</tr>
</tbody>
</table>

Table 2 shows the percentage of respondents selecting each style as their favorite out of 84 respondents who answered this question. Similar to our other evaluations, we see that no one style is dominantly favored.

Table 2: Percentage of respondents choosing style as favorite

<table>
<thead>
<tr>
<th>Style</th>
<th>% Favorite</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autoencoder (A6)</td>
<td>14%</td>
</tr>
<tr>
<td>Carpet (A3)</td>
<td>19%</td>
</tr>
<tr>
<td>Circle Packing (A1)</td>
<td>23%</td>
</tr>
<tr>
<td>Glass (A5)</td>
<td>8%</td>
</tr>
<tr>
<td>Tile (A4)</td>
<td>25%</td>
</tr>
<tr>
<td>String Doll (A2)</td>
<td>11%</td>
</tr>
</tbody>
</table>

Across our evaluations, we see that (1) a majority of subjects find our visual representation of abstract concepts intuitive, (2) our NLP model extracts accurate labels for a majority of entries, (3) a majority of subjects prefer our motifs...
over corresponding baselines, and (4) a majority of prospective users consider Lemotif a useful system that can increase their likelihood to journal and enjoyment of journaling.

Future Work

Future work includes developing Lemotif into an app allowing users to track entries over time and view temporal (e.g. weekly, monthly) summaries. The app could allow for custom mappings, such as allowing users to specify the name of their job so the NLP model always identifies it as “work,” or correct the detected topics and emotions so that over time the model learns the user’s personal life and writing style. Training our model with more data containing more diverse labels would also likely improve its performance.

Additional visualization styles are possible given the diversity of generative art. Our autoencoder model would likely improve with architectural changes, adversarial discriminator loss (like a GAN), and hyperparameter tuning. With a sufficiently large and annotated dataset, a conditional GAN could be trained that takes in color labels directly rather than as low-resolution images. Multiple models could be trained on different artists and artistic movements. Within an app system, users could provide feedback on generated motifs they like more or less, further training the image models to the user’s own taste. Additional input dimensions like the intensity of emotion could be incorporated, such that stronger emotions appear more saturated.

Conclusion

In summary, we present Lemotif. It takes as input a text-based journal entry indicating what aspects of the user’s day were salient and how they made them feel and generates as output a motif – a creative abstract visual depiction – of the user’s day. As a visual journal used over periods of time, Lemotif aims to make associations between feelings and parts of a user’s life more apparent, presenting opportunities to take actions towards improved emotional well being.

Lemotif is built on five underlying principles: (1) separate out the sources of emotions, (2) depict these sources visually, (3) depict these emotions visually, (4) generate visualizations that are creative and attractive, and (5) identify and visualize detected topics and emotions automatically using machine learning and computational methods. We verify via human studies that each of the first four factors contributes to the proposed motifs being favored over corresponding baselines; accuracy and F1 metrics indicate the NLP model greatly outperforms random chance. We also find that subjects are interested in using an app like Lemotif and consider the generated motifs representative of their journal entries.

Acknowledgments

Thanks to Ayush Shrivastava, Gauri Shri Anantha, Abhishek Das, Amip Shah, Sanyam Agarwal, Eakta Jain, and Geeta Shroff for participating in the study. Special thanks to Abhishek Das for useful discussions and feedback. At Facebook AI Research, we understand that researching a topic like emotion is nuanced and complicated. This work does not research what causes emotional well being (or not). It does not use Facebook data or the Facebook platform in any way. It simply generates visualizations based on topics and emotions reported by subjects explicitly electing to participate in our study, and analyzes which visualizations subjects prefer. Creative applications of AI are a powerful avenue by which AI can collaborate with humans for positive experiences. This work is one (small) step in that direction.

References

Trubetskov, S. 2017. List of 20 simple, distinct colors.
3D Topology Transformation with Generative Adversarial Networks

Luca Stornaiuolo, Nima Dehmamy, Albert-László Barabási, and Mauro Martino

1 Politecnico di Milano, 2 Northeastern University, and 3 IBM AI Research
luca.stornaiuolo@polimi.it, {n.dehmamy, a.barabasi}@northeastern.edu, mmartino@us.ibm.com

Figure 1: Our proposed 3D Topology Transformation method disentangles the concepts of structural shape and volumetric topology style of objects in 3D space. The Generator of our Vox2Vox is able to transform the 3D input models (on the top) to novel 3D representations (on the bottom), while retaining the original overall structure.

Abstract

Generation and transformation of images and videos using artificial intelligence have flourished over the past few years. Yet, there are only a few works aiming to produce creative 3D shapes, such as sculptures. Here we show a novel 3D-to-3D topology transformation method using Generative Adversarial Networks (GAN). We use a modified pix2pix GAN, which we call Vox2Vox, to transform the volumetric style of a 3D object while retaining the original object shape. In particular, we show how to transform 3D models into two new volumetric topologies - the 3D Network and the Ghirigoro. We describe how to use our approach to construct customized 3D representations. We believe that the generated 3D shapes are novel and inspirational. Finally, we compare the results between our approach and a baseline algorithm that directly converts the 3D shapes, without using our GAN.

Introduction

Artificial Intelligence (AI) is experiencing enormous growth in popularity within the fields of creative arts and design. Image style-transfer based on deep neural networks (Gatys, Ecker, and Bethge 2016; Zhu et al. 2017) has become very popular both in scientific and artistic communities, influencing image creation as well as videos and movies production (Huang et al. 2017). This approach allows achieving a decoupling of content and style in art (Johnson, Alahi, and Fei-Fei 2016), where AI has become able to generate new pieces of art using the style of one piece and the content of another.

In the context of utilizing machine learning algorithms to produce new artworks, Generative Adversarial Networks (GAN) have played an integral role in the recent studies, owing to their ability to learn representations of data and to generate outputs that mimic realistic elements (Radford, Metz, and Chintala 2015), including songs, paintings, and sketches (Briot, Hadjeres, and Pachet 2017; Dumoulin, Shlens, and Kudlur 2017; Ha and Eck 2017). While in many GANs the input is a random noise, conditional GANs (cGANs) (Mirza and Osindero 2014) can be used to pair a specific input with the desired generated output, as in the case of the pix2pix architecture (Isola et al. 2017), able to generate new realistic images from different representations. While most of these efforts in using GANs have focused on 2D
images, the technology has become mature enough to apply these methods to 3D shapes. Multiple works have generated realistic 3D reconstruction of shapes starting from photos (Wu et al. 2016; Brock et al. 2016) Recently, there have also been many works on generating or learning a point cloud representation of 3D objects (Achlioptas et al. 2017; Li et al. 2018). There have also been many works on converting 2D images or depth-maps to 3D (Li et al. 2019; Shin, Fowlkes, and Hoiem 2018). However, there has been little research on how to create stylistically different 3D representations of a given 3D shape.

In this paper, we present Vox2Vox, a new 3D cGAN, that generalizes pix2pix to 3D. As shown in Figure 1, Vox2Vox is capable of transforming the topology and the internal structure of a 3D object, while maintaining its overall shape. We propose two volumetric topologies, the Ghirigoro and the 3D Network, and we describe how to train our Vox2Vox model to create new topology transfers. We then demonstrate the effectiveness of our approach by comparing the results obtained with our AI model against those obtained with a pure procedural algorithm.

### Related Work

Before proceeding to describe the details of our approach, we will review other works on creating 3D shapes with AI.

**3D Shapes Synthesis.** 3D shape synthesis is used to model and reproduce shapes in 3D space. While the first applications of 3D geometry models mainly concerned video games and visual media, nowadays, reproducing 3D shapes of real world objects influences design and architecture, to self-driving cars, to scientific and medical visualizations. Our work contributes to a shift from the manual creation of 3D models with computer-aided design tools to automatically generated 3D shapes from different inputs. In addition to approaches that use probabilistic graphical models (Kalogerakis et al. 2012), the advent of Deep Learning brings the opportunity to automatically synthesize novel 3D shapes by assembling parts of 3D objects, extracted from model databases, to create new compositions (Huang, Kalogerakis, and Marlin 2015). However, these approaches require collections of labeled 3D object parts, which despite the release of many 3D databases are still hard to acquire. To relax the labels requirement, view-based generative models are employed. They allow for reconstruction of 3D shapes given 2D images that represent one or more view points of the object (Su et al. 2015; Qi et al. 2016; Soltani et al. 2017). However, reconstructing 3D shape starting from its 2D representation often yields low quality results due to the missing information, making it hard to produce unique inputs and realistic 3D objects.

**Generative Adversarial Networks.** GANs have become one of the most popular neural network architectures to generate novel realistic outputs (Radford, Metz, and Chintala 2015; Brock, Donahue, and Simonyan 2018). GANs are composed of two neural networks, the Discriminator, trained to distinguish between the real and the generated inputs, and the Generator, trained to produce new outputs that look real. GANs have been used to generate voxel-based object representations in the 3D space (Brock et al. 2016; Wu et al. 2016), starting from random noises. However, producing a desired shape requires exploring the latent space in order to find the correct random input for the Generator. In this context, cGANs introduce non-random inputs to control the outputs of the Generator (Goodfellow et al. 2014). cGANs have been employed as 3D shape Generators in (Wu et al. 2016; Xie et al. 2018), where 2D images and a conditional probability density function are respectively used to condition the output of the Generator and map the input to the desired 3D shapes. However, only few works exist about controlling the output of the Generator with 3D shapes as conditional inputs. (Ongun and Temizel 2018) presents a 3D-cGAN able to perform rotations of volumes in the 3D space. In comparison, our approach aims to transfer the shape of the input, while modifying its volumetric topology to obtain a novel collection of 3D objects, and, to the best of our knowledge, our is the first 3D-cGAN model for 3D topology transformations.

**Style Transfer.** Style transfer aims to learn the content and the structure of an input element while changing its style based on a different style source. In the past, many works have been released that perform 2D-to-2D style transfer on images and videos (Johnson, Alahi, and Fei-Fei 2016; Gatys, Ecker, and Bethge 2016). 2D-to-3D style transfer is mainly employed to apply specific textures or colors from 2D samples to 3D objects (Nguyen et al. 2012). (Kato, Ushiku, and Harada 2018) presents a novel 2D-to-3D style transfer approach able to perform gradient-based 3D mesh editing operations to modify also the surface of the 3D shapes based on the image used as style source. (Ma et al. 2014) is one of the first works about 3D-to-3D style transfer: the proposed algorithm computes the analogy between one source element and the related target and applies it to synthesize new outputs based on different sources. This deformation-based approach is also used to generate different poses of animal meshes (Sumner and Popovic 2004) and modify the design of 3D furniture, buildings models, and different classes of objects (Xu et al. 2010; Liu et al. 2015; Lun, Kalogerakis, and Sheffer 2015; Mazeika and Whitehead 2018). However, these works require a specific formulation of analogy between the different parts of the analyzed objects that limits their application to few collections of 3D models. In this context, our work lays the foundation for a novel 3D-to-3D style transfer. Indeed, with our 3D-cGAN model for 3D topology transformations, we can say that the style is sedimented in the trained network and multiple styles are supported with different weights of our Vox2Vox Generator. A future step would be allowing both shapes and styles as inputs of the Generator.

### Approach

The goal of this paper is to convert a 3D shape into an alternative 3D representation inspired by the original. We will focus on converting the shape into a 3D Network, which is an abstract representation in the same vein as cubism in art. A network (graph) consists of two types of components:
Figure 2: Pipeline. We present two 3D representations: the 3D Network topology (top) and the Ghirigoro topology (bottom). Our pipeline starts with a 3D model and converts it to its filled 3D voxel representation. The 3D voxel representation is fed as input to the Generators of Vox2Vox trained to perform the topology transfers. The results are the new 3D voxel representations, where different channels contain different information of the new 3D shape (e.g. node and link distributions and for the 3D Network topology). Finally, the output 3D model is reconstructed from the voxel representation with our Procedural Network algorithms.

1) Nodes, being the entities to be connected and 2) Links, which are the wires connecting the nodes. Like scaffold that supports the interior of the 3D object, we want the nodes to be placed in suitable locations inside the shape and links to connect these nodes to form a 3D Network.

However, as we will discuss later, a random or a uniform choice of where to position nodes and link points doesn’t result in aesthetically pleasing results. Thus, we have to find a better way to choose where to place the nodes and link points. This requires us to find what part of the 3D shape should contain more nodes in order for the shape to be represented better than random nodes, and we would have to hand-code the criterion for finding this node density. If, however, the aesthetics is changed slightly, e.g. having curvy links like a scribble taking the shape of the network, we will need to, again, hand-code a procedure for the new aesthetics. Solving this inverse-problem of finding an algorithm or rule for suitable distribution of node and link positions inside the 3D shape can be very difficult. In contrast, training a system that would learn by examples could be much easier. Therefore, we want our Vox2Vox cGAN to be able to produce the node and link distributions, using voxels. Once we have this distribution, we can use a procedural algorithm, called Procedural Network, that takes the voxels as an input (together with the number of nodes and the number of points to be used for links connecting nodes) and generates the final 3D Network representation.

In this way the main challenge moves from creating an algorithm that performs the transformation of the desired topology (in this case Network 3D) to creating the dataset necessary for training the AI model. Fortunately, creating arbitrary 3D Networks and converting them to space-filling 3D blobs is much easier than doing the reverse, as we will discuss in the section about model training. Once we have 3D Networks with their respective 3D figures, the 3D-cGAN is able to learn how to convert the latest to the former, while finding a pure procedural algorithm for doing so may be difficult and an optimization process may be costly. Another benefit of this approach is that, if we want to choose a different aesthetics, we can create a new aesthetics-related training dataset and not worry about the inverse problem.

Proposed Method
As stated in the previous section, solving the problem of creating a neural network to perform topology transfer is equivalent to creating a neural network that can take a 3D shape
and produce a voxels-based distribution of nodes and links to put inside it. Thus, the output should preserve the general 3D structure of the input. It was shown in (Isola et al. 2017) that preserving the overall location of features in the input and output is important, the U-Net architecture performs much better than an encoder-decoder architecture, owing to its skip layers. U-Net (Ronneberger, Fischer, and Brox 2015) is also known to converge with relatively little training data, making it ideal for our application. We, therefore, choose a U-Net architecture for the main component of our neural network. However, given a 3D shape, there are many different ways that one can fill this shape with a network. This means that when generating 3D blobs from networks, many different network layouts may end up having similar 3D blobs. This will confuse a neural network, as the same input is being assigned different 3D Networks as “label”. This is precisely where a cGAN becomes useful. Unlike a simple neural network, where multiple labels would result in the network choosing the average of those labels, in a GAN the loss is minimized as long as the GAN learns to produce one of the correct labels, so that the Discriminator cannot tell the output from the real data. In summary, we need a cGAN which has U-Net architecture for its Generator and a Discriminator suitable for classifying 3D shapes. This is essentially exactly what the pix2pix architecture (Isola et al. 2017) does, only in 3D instead of 2D. Thus, the architecture of Vox2Vox will be very similar to pix2pix, but the 2D convolutional layer will be replaced with 3D convolutions and the number of layers and filters will be different.

Pipeline Overview. Figure 2 shows an overview of the pipeline for converting a 3D shape into a 3D Network. First we transform a 3D input mesh to a filled voxel representation, given a target 3D resolution. The filled voxels are passed to the Vox2Vox Generator which outputs two channels of voxels: ch. 1 for the distribution of where to put nodes and ch. 2 for distribution of points along links connecting the nodes. Finally, the voxel distributions for nodes and links are passed to the Procedural Network (described in detail below) algorithm to produce the final 3D Network.

Model Architecture. Table 1 shows the details of the modules used in the Vox2Vox Generator and Discriminator. The Generator has a U-Net architecture, with four Down3D modules in the encoder and four Up3D modules in the decoder each of which contains a 3D convolutional layer. During training, the input shape is $64 \times 64 \times 64$ and the encoder layers have 32, 64, 128 and 128 filters, encoding the input to $4 \times 4 \times 4 \times 128$ shape. The decoder layers have 128, 64, 32 and $C$ filters. $C$ is the number of desired output channels, which for the 3D Network problem is two, one for node and one for link distribution. The Up3D modules also get the output of the encoding layers as input, as in U-Net. The Discriminator consists of four 3D convolutional layers, yielding a $8 \times 8 \times 8$ output. The cost function for the Discriminator decides whether on all the $8 \times 8 \times 8$ patches of the node and link distributions match the input 3D blob in a certain way. If the Generator successfully produces a node and link distribution which matches the blob over all the $8 \times 8 \times 8$ similar to how the real data matches the blob, the Discriminator will get fooled. Because of the third dimension, the training process of Vox2Vox is both very memory intensive and computationally very expensive. That is why we chose to do the training in the resolution of $64 \times 64 \times 64$ voxels. However, during prediction we increase the input size to produce higher quality outputs. Indeed, a nice feature of U-Net is that, since all layers are convolutional, we can produce larger outputs by simply modifying the input shape of the trained network. Note that, this increase in input size will not result in larger “features”. For example, if the Vox2Vox trained at resolution 64 produces nodes with radius 5 voxels maximum, increasing the input shape to 192 will still only produce nodes of radius 5. Similarly, if at resolution 64 the maximum length of links produced is 10 voxels, it will be the same when the input size is changed to 192. This is because the convolutional layers in all layers will still have the same receptive field of, say $8 \times 8 \times 8$ on the input, at both 64 and 192 resolutions. However, what this allowS us to do is that we can feed arbitrary large inputs to a trained Vox2Vox (192 was the maximum size our GPU memory allowed) and convert them into 3D Networks. Moreover, making the input larger allows us to convert objects with much higher details than the size of the objects in the training set. Therefore, after we train Vox2Vox on $64 \times 64 \times 64$ voxels, we change the input shape to $192 \times 192 \times 192$ and produce higher quality outputs. We used Vox2Vox on a variety of different 3D sculptures and passed the output to the Procedural Network Algorithm to extract the generated 3D Network (Fig. 4).

**Table 1: Model Architecture.** The Vox2Vox Generator has a U-Net architecture made of Down3D modules which contain 3D convolutional layers, and Up3D modules which does an Upsampling by a factor of 2 in each direction, followed by a 3D convolutional layer, Dropout and Batch Normalization. The Discriminator consists only of Down3D modules. The number of filters, output dimensions and the number of parameters for each layer in the Generator and Discriminator are reported.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Output shape</th>
<th>Filters</th>
<th>Params</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generator</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input Layer</td>
<td>$64^3$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Down3D</td>
<td>$32^3$</td>
<td>96</td>
<td>32,080</td>
</tr>
<tr>
<td>Down3D</td>
<td>$16^3$</td>
<td>48</td>
<td>131,392</td>
</tr>
<tr>
<td>Down3D</td>
<td>$8^3$</td>
<td>24</td>
<td>524,928</td>
</tr>
<tr>
<td>Down3D</td>
<td>$4^3$</td>
<td>12</td>
<td>2,048,216</td>
</tr>
<tr>
<td>Up3D</td>
<td>$8^3$</td>
<td>24</td>
<td>1,049,216</td>
</tr>
<tr>
<td>Up3D</td>
<td>$16^3$</td>
<td>48</td>
<td>1,048,896</td>
</tr>
<tr>
<td>Up3D</td>
<td>$32^3$</td>
<td>96</td>
<td>262,304</td>
</tr>
<tr>
<td>Up3D*</td>
<td>$64^3$</td>
<td>$192^3$</td>
<td>8,194</td>
</tr>
<tr>
<td>Discriminator</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input Layer</td>
<td>$64^3$</td>
<td>$C$</td>
<td></td>
</tr>
<tr>
<td>Down3D</td>
<td>$32^3$</td>
<td>1</td>
<td>3,088</td>
</tr>
<tr>
<td>Down3D</td>
<td>$16^3$</td>
<td>3</td>
<td>32,080</td>
</tr>
<tr>
<td>Down3D</td>
<td>$8^3$</td>
<td>64</td>
<td>131,136</td>
</tr>
<tr>
<td>Down3D*</td>
<td>$8^3$</td>
<td>1</td>
<td>32,769</td>
</tr>
</tbody>
</table>

* No Batch Normalization + Kernel size=8
Figure 3: Model Training. Sequence of 3D Network outputs from the Vox2Vox Generators while growing of epochs.

Figure 4: Training Vox2Vox: For conversion to a 3D Network, we first create a network and lay it out in 3D. Then we convert the network into two formats: 1) a 3D blob found by assigning large radii to nodes and large thickness for links and merging them together (input of the Generator); 2) Separate node and link distribution data with nodes having reasonably small radii and links being relatively thin (real input of the Discriminator). The output of the Generator will have two channels, the predicted node and link distributions (fake input of the Discriminator).

inputs, which is the concatenation of the $64 \times 64 \times 64$ array of the 3D shape with the $64 \times 64 \times 64 \times C$ array of real or fake output along the channels, yielding a $64 \times 64 \times 64 \times C + 1$ array. For the real pair, the label is a $4 \times 4 \times 4$ array of ones, and for the fake pair the array is zero. The main difference is our training data is 3D and completely synthetic (see sections about training data below). Some points to note is that, we exploited the convolutional nature of the U-Net layers also during training. During the training process, the input is a $64 \times 64 \times 64$ binary array (0 in empty spaces and 1 inside the 3D shape) and output of the Generator is a $64 \times 64 \times 64 \times C$ array, where $C$ is the number of channels. While this training converges to reasonably good results after around 10 epochs, we found that first training on $32 \times 32 \times 32$ binary arrays and very small networks, and then doing transfer learning by increasing the input resolution resulted in faster convergence to reasonably good results.

3D Network Training Data. We create a variety of networks first and then lay them out in 3D using a custom forcedirected layout algorithm, inspired from (Dehmamy, Milanlouei, and Barabási 2018), which makes sure nodes do not overlap. We then create two arrays from the network. One is a $64 \times 64 \times 64$ array of the 3D shape of the layout, merging nodes and link segments by replacing them with overlapping large and small spheres, respectively. This 3D shape is the input of the Generator. The second is a $64 \times 64 \times 64 \times 2$ array, with the first channel being the nodes and the second the links, which are again replaced by spheres, this time with smaller radii (Figure 4). Then, we replace nodes with spheres and segments along links with spheres which overlap and make the connected links. We generate about 700 such networks and heavily augment the dataset by rotating them in multiples of 20 degrees about the x,y and z axes to create a dataset of about 30,000 data points. We trained Vox2Vox with a few different thicknesses for links and sizes for nodes to determine a good choice for the sizes of the nodes and links so as to avoid space-filling links and nodes, which would let the Generator exploit this structure and fool the discriminator by simply filling the inside of the shape. Figure 3 shows a sequence of 3D Network outputs from the Vox2Vox Generators while growing of epochs.

Note on Input Network Topology. There exist many different generative processes for producing networks which result in very different connectivity patterns. We found that using random graphs (Erdős and Rényi 1960) resulted in poor results. In these graphs, any node is equally likely to connect to any other node, resulting in a completely random network, with all nodes having a similar degree (i.e. number of links attached to them). The Vox2Vox Generator was able to produce reasonable positions for the nodes, but failed at producing good links between them. Our hypothesis is that, since links in these networks are completely random, the Discriminator overfitted to the few samples it had been trained on and was never satisfied with any other variation of the links. In contrast, when we trained the Vox2Vox on networks generated using a “rich gets richer” (Barabási-Albert (BA)) model (Barabási and Albert 1999) the results improved dramatically. In the BA model some nodes, known as hubs, have significantly more links than other nodes. The BA model has “hubs” which are nodes with a much higher degree than most other nodes, which contrasts it strongly from the random ER network where all nodes have more or less the same degree. We chose the size of the nodes in the 3D array to be a function of their degree. When making the voxel representation, we assign larger node radii nodes with a higher degree. We believe that this predictable correlation between the node size and the density of links was learned by the Generator and was exploited to fool the Discriminator, allowing it to produce good results.

Usually, to train GANs a single input is presented to the network at a time (Goodfellow et al. 2014), meaning batch size 1. This will force the Generator to try to learn the features of a single example. However, this will result in a very slow progression of the training. On the other hand, larger
batch size yields faster convergence, but presenting a large batch of data result in more of an averaged output, stopping it from choosing a single pattern. Therefore, in the initial stages of the training, we set the batch size to 8 to allow for a faster approach towards good filters. In later stages, we reduced the batch size to 2 and then 1 to fine-tune the results.

**Ghirigoro: Training on a Second Topology.** Aside from converting shapes to 3D Networks, we also tried a second, related topology. The “Ghirigoro” (doodle) topology consisted of converting a 3D shape into a long scribble that mimics its shape. This style has only one output channel, which is a set of long curvy lines, crossing more rarely than the 3D Network case, and there are no nodes. To construct the final doodle shape from the GAN we used the Procedural Network algorithm with different settings. The result is not exactly a doodle, as the Procedural Network may cross-link two parallel pieces of the doodle, resulting in a network. Nevertheless, the aesthetic is close to a doodle. The pipeline is shown at the bottom of Figure 2 and the results are presented along with the 3D Network case in Figure 6.

**Procedural Network algorithm**

We interpret the two output channels of the Vox2Vox Generator as node and link distributions. To extract a network from these distributions, we first use K-means to choose centers for nodes and for points along the links. We choose a higher K for link points than nodes in the K-means. We then form a network from points that are close to each other, connecting link points and node centers in a dense network. To avoid connecting points that are far from each other, we partition the space into small regions and connect points within each partition to each other. To avoid artifacts from the position of partition walls, we repartition the space multiple times by slightly shifting the partition walls by a random value. This method of connecting nodes and link points works better than a k-nearest neighbor method, as all nodes need not have the same degree, and hence neighbors. This results in a series of connected paths connecting nodes using via intermediate link points. Finally, we use the A* search algorithm (Hart, Nilsson, and Raphael 1968) to extract links between nodes.

To see how using the Vox2Vox output as a prior for node and link distribution compares against not using having this prior, we apply the above procedure to the original 3D shape voxels. This means that, to find node locations we just take the filled voxels of the 3D shape and apply K-means to it. This will fill the interior of the shape with randomly chosen node locations. Our default was choosing 300 node locations. We then do the same thing for link points, except that we choose 50 times more points to serve as points along
We show the results of the proposed 3D Topology Style Transfer approach. For each 3D input model (first column from left), we present its volumetric transformations based on the 3D Network topology (second column) and the Ghirigoro topology (third column). We then compare the results of the Procedural Network algorithm (fourth column) that applies the volumetric transformation directly on the 3D input.

links (i.e. $50 \times 300$ points). The rest of the procedure is the same. Comparing the outcomes, we observe that the baseline procedure results in curved and broken links, while the proposed Vox2Vox model produces 3D shapes with more effectively distributed links and nodes.

Results
Figure 5 shows the results of the proposed 3D topology style transfer approach. For each 3D input model, we present its volumetric transformations into the 3D Network topology and the Ghirigoro topology. We then compare the results of the Procedural Network algorithm (fourth column) that applies the volumetric transformation directly on the 3D input.

In this paper, we presented a novel 3D-to-3D topology transfer paradigm based on transformations in 3D space. In particular, we built a 3D conditional GAN, Vox2Vox, that performs volumetric transformations to modify the internal structure of any 3D object, while maintaining its overall shape. We described our complete pipeline to apply our approach to two different topologies: the 3D Network and the Ghirigoro. The results obtained by employing our methodology are novel and inspirational. We compared the outputs of the pipeline while using or not the 3D-cGAN and found that using the Vox2Vox output as a prior distribution results in much nicer outcomes where features are placed in strategic positions in the 3D shape preserving its structural features. As a future direction, we plan to improve the 3D-to-3D topology transfer by given also the topology as a conditional input of the generative network. To do that, the machine learning algorithm has to learn itself the abstraction of the topology from a given 3D object.

Conclusion and Future Direction

In conclusion, we achieve a novel 3D-to-3D topology transfer paradigm based on transformations in 3D space. In particular, we build a 3D conditional GAN, Vox2Vox, that performs volumetric transformations to modify the internal structure of any 3D object, while maintaining its overall shape. We described our complete pipeline to apply our approach to two different topologies: the 3D Network and the Ghirigoro. The results obtained by employing our methodology are novel and inspirational. We compared the outputs of the pipeline while using or not the 3D-cGAN and found that using the Vox2Vox output as a prior distribution results in much nicer outcomes where features are placed in strategic positions in the 3D shape preserving its structural features. As a future direction, we plan to improve the 3D-to-3D topology transfer by given also the topology as a conditional input of the generative network. To do that, the machine learning algorithm has to learn itself the abstraction of the topology from a given 3D object.

References


Hello, An Interactive Cinematic Generative Artwork

Pedro Alves da Veiga
Centro de Investigação em Artes e Comunicação
Universidade Aberta, Universidade do Algarve
Portugal
me@pedroveiga.com

Abstract
“Hello” is an interactive cinematic generative artwork that gained traction within the Cultural Adventures research project, an international endeavor between Portugal and Thailand. The encompassing rationale assumed the need of a link connecting both countries, and thus “Hello” was born from the similarities that can be found in commuters that flock to and from both capital cities by train, often more engaged with their mobile phones than with their immediate surroundings — people, train or traversed landscapes —, which constitute more of a shapeless blurred audiovisual background than a sharp attention-grabbing focus. The artwork is also used as a demonstrator of the potential of generative art in producing complex cinematic narrative figural artworks, yet permeated with interference, chaos and randomness, and how the audience perceives and makes sense of them through the ability to identify and create patterns, to conceptualize, to deal with information overload, and to imagine and build a narrative through abductive reasoning from incomplete yet suggestive information.

Introduction
Common human everyday visual perception is pragmatic, and is mostly oriented toward the identification of objects and shapes in visual scenes. Individuals report perceiving objects in pictorial compositions even when those compositions are devoid of recognizable objects (Ishai, Fairhall and Pepperell 2007), which points to an inescapable human trait: that of searching for patterns.

Inspired by the work of Ezra Pound, McDonald (1993) proposed that by opening up the perceptual field to the indeterminacies of fragment and chaos, it became possible to recreate a multi-sensory field within which audience and creator encounter the same stimuli toward a desired end. This process enables the role of quester (after patterns, knowledge or meaning) to be held simultaneously or passed around audience members, who need to chart an unknown, unstructured territory, collecting partial findings as the exploration takes place in real time. The absence of (expected) markers unsettles narrative assumptions and challenges the audience from the comforts of convention and its passive position into an experience of navigation by peripus. The audience is no longer outside the artwork and its narrative, but inside it, mapping its shape and meaning as they gather information and act upon it.

The term generative implies an algorithmic framework within which takes place the creation of the artwork itself. The algorithm combines structure with controlled randomness, bringing order and chaos together, in line with Pond’s multi-sensory field, and each iteration becomes the seed for the next iteration, thus resulting in a seemingly infinite, non-repetitive variety of states within a certain aesthetic boundary defined by the artist / programmer (Galanter 2014).

“Hello” is an interactive cinematic generative installation that outputs a constant stream of images and sound, which can, to some extent, be influenced by the audience. The installation foresees two different audience behaviors as it is regarded both as a chaos generator and as a system exciter. If no — or little — movement is detected in the exhibition room, the (still) audience is presented with a continuous stream of a flickering mosaic set to a background track of ambient train-related noises. All the images used to produce this stream are of Thai people, landscapes, trains, platforms and stations. All the sounds were captured in Portuguese trains and train stations. This sets the mood to an imaginary journey, where images are deconstructed and reconstructed, much like what happens in a commuter’s mind, traversed by the fleeting landscapes, glimpses of their fellow travellers and of the their own imagination, triggered by thoughts, conversations and other stimuli. However, a curious and more dynamic audience will be presented with a stream of short text messages, half of which are in Thai (left side of the projection) and the other half in Portuguese (right side of the projection), visually inspired by most chat and messaging systems currently in use and replicating their expected behavior of moving upward as new utterances are introduced into the dialogue. This dialogue becomes more active as the artwork detects more movement in the exhibition room, which also impacts the several parameters of the mosaic generation and sound intensity, as the installation echoes the action in the exhibition room. An added element of surprise is brought to the equation as the system captures random video frames of the audience and introduces them among the
original Thai imagery, literally bringing the audience into the artwork.

Figure 1 shows a run-time screen-capture of “Hello”, with Thai (left) and Portuguese (right) dialogues underway.

Figure 1 - Screen capture of "Hello", with Thai and Portuguese dialogues underway, over a cinematic background.

The dialogues in “Hello” contain hints of a love-crime fiction, spanned and scattered across multiple utterances from both interlocutors. As the audience realizes that the more movement is detected in the exhibition room, the more utterances are provided, thus allowing them to access more information and complete the narrative, it is expected that the exhibition room then may become the stage of an audience performance, both physical and intellectual, as members of the audience may engage in collaborative speculation and debate.

A video with a real-time capture of "Hello" is available at https://vimeo.com/alvesdaveiga/hello.

Methodology
Due to its nature, this project – both the artwork development and the related research – were brought to term using a/r/tography (Veiga 2019), an arts-based research methodology derived from a/r/tography (Springgay, Irwin and Kind 2005). This methodology stipulates seven generative, mutually influencing steps and its outcome involves the artwork, the documented research, the communication of any project related outcomes, including public exhibition and presentations, and all the inherent processes. These steps are presented over the next sections.

Inspiration
The concepts of communication and transportation have some interesting similarities and intersections: trains connect distant locations and mobile devices connect distant people; trains carry our physical presence, while mobile devices carry our immaterial presence. But a train track also divides territories, in the same way that a mobile device can separate people who share a common location, by allowing them to easily disconnect from their physical surroundings and engage with a remote (non-local) physical or virtual technology-mediated reality. This was rendered evident through observations in public transportation in the greater Lisbon area, in Portugal, where local commuters are often absorbed in the use of their devices rather than in conversations with nearby fellow travellers or showing an interest in the visual details around them, both inside and outside the train.

Trigger
Similarities in this type of behavior were highlighted during conversations with Phanuphong Songkhong, a Thai designer, who documented very similar observations in several thousand drawings he produced during his own commutes, in and out of Bangkok.

Acknowledging so many similarities in distant countries, with such different cultures, was the trigger that set in motion the whole project.

Intention
From the original inspiration stemmed the will to translate into an artwork how the whole sensorial stimulation during such train journeys provided a mosaic of mixed sensations in the brain, some of which were the direct result of vision and hearing, others the mental offspring of the activities people engaged in (texting, e-mailing or telephone conversations), and the rapidly changing mosaic-like frame of mind that ensued.

There was also the intent to portray a reality shared by Portuguese and Thai daily commuters, where technology mediated dialogues were apparently more easily established than local dialogues, as the majority of regular commuters were usually more deeply engaged in interactions with their mobile devices, rather than having conversations with their travel companions.

All these factors contributed to the intention of delivering a mosaic-like atmosphere, where stimuli are segmented and blurred into an all-encircling and all-absorbing yet challenging experience, without requiring the use of virtual or augmented reality to accomplish its immersive effect – much like the situations it depicts.

Conceptualization
All the overlapping, mosaic-like stimuli found in a train commute contribute to creating a sensory journey of fleeting visions and sounds, driven by our immediate physical surroundings, by conversation-triggered memories or imagined scenes, or even our own personal ramblings and self-awareness.

Replicating this sensory stream will invite the audience to embark on a cross-continental immersive and interpretive journey. As visitors enter and move in the exhibition room, they are already unknowingly interacting with the installation, potentially becoming “the subject of conversation”, as images of the room are captured and inserted into the visual stream, and the “journey conductor”, because their movements in the room directly affect system behavior.
As the audience is presented with a dialogue, of which (probably) only half is intelligible, they are also invited to imagine and reconstruct a narrative through abductive reasoning from incomplete yet suggestive information (Peirce 2012).

Related work
Generative Cinema may not be a widespread genre but has a long history rooted in earlier forms, among which is Calculated Cinema. For Bonet (2007) the Calculated Cinema series intends to open a window of observation allowing a glance both backwards in time, as well as forwards, offering "an antidote for the digital animation styles which are now hegemonic and omnipresent" (Bonet 2007).

In Calculated Cinema, directors such as Sergei Eisenstein and Dziga Vertov laid the foundations of metric or rhythmic montage, adopted in the current artwork. Manovich and Kratky (2005) consider this approach to be at the heart of Soft Cinema: if hard cinema is based upon storylines, scripting, shooting and editing guidelines, Soft Cinema is based upon algorithmic sequences of visual units extracted from a previously constructed database, often including randomness in their selection mechanisms. Soft Cinema advocates the replacement of single-screen images by mosaics of images. Each new screening will differ, due to the randomization of the database units and large number of possible combinations.

Soft Cinema and Calculated Cinema are thus both at the heart of Generative Cinema, which specialises in the delivery modes of a heterogeneous realm of artistic outcomes based upon the combination of predefined elements (order) with different factors of unpredictability (chaos) in its conceptualization, production and presentation (Grba 2017, 8).

In "Hello" generative processes become richer and more complex through the introduction of probabilistic and random factors in predetermined rules, thus producing successive different generations of audio-visual output where each newer generation inherits structural and compositional characteristics of the previous one(s), as well as randomness and non-linearity.

Prototyping
The idea to portray such a multisensory mosaic took shape and the project was founded around three key areas: movement detection, dialogue generation and a cinematic effect obtained from still images.

Movement detection
The interaction is built upon the analysis of consecutive frames, captured by a webcam facing the audience. This analysis takes into consideration noise levels – higher noise levels in darker environments, lower levels in well-lit environments – and a pixel-by-pixel image differential (Singla 2014; Suresh and Lavanya 2014):

$$\Delta I_{(i,j)} = I_{Current}(i,j) - I_{Prev}(i,j)$$

This allows for the determination of the general area of the room where most movement occurs, i.e. where most different pixels between consecutive frames are found. Since the purpose of movement detection in this artwork is merely to determine the prevalence of utterances on the Portuguese side (right) or the Thai side (left) of the projection, its calculation is simple enough to be undertaken in real-time with no hindrances to the overall artwork performativity.

Dialogue generation
The dialogue is generated from a database of utterances, some of which – Portuguese – were obtained from the author's own SMS conversations, while others –Thai – were provided by Thai students from King Mongkut’s University of Technology Thonburi, in Thailand. These utterances were later complemented by specially created content in both languages, in order to weave the underlying plot.

Since the text-based dialogue is presented in those two seldom-learned together languages, it is fair to expect that no audience member will know both of them, and here lies the heart of the game: the purpose of any dialogue is to exchange information between two interlocutors. When presented with only one intelligible information source, the audience will gather key segments of information about a certain topic from different utterances, some of which are meaningful (e.g.: “He robbed her?”; “She fooled everyone”, “It was a red house”) while others are conversation fillers (e.g.: “I see”, “Is that so?”) leading the audience into believing that the key information was on the other (unintelligible) part of the dialogue. A total number of 400 utterances were produced in both languages.

The presentation of different utterances follows the established chat-room paradigm that determines newer messages should appear at the bottom of the screen and push older messages upwards. The system was thus implemented with a FIFO message queue for each language, storing the screen coordinates for every utterance. As the screen coordinates are updated, if one utterance exits the screen (negative y coordinate) then it is deleted from the queue. Each utterance has a time-to-live (TTL) property, which determines that older messages will be discarded after their life-span reaches zero. The purpose of this mechanism is to clear the projection space of all messages if no movement is detected for a certain amount of time – corresponding to a group of visitors having exited the room or being very still for very long. If the queue is empty, then the first message that will next be presented in either language is “Hello” (Olá, สวัสดี).

Each time an utterance is displayed, it is marked as used (Boolean state), to avoid repetition of previously and recently used utterances. After all utterances have been used, they are marked as unused and the process restarts.

A compensation mechanism was also implemented to prevent messages in Portuguese or Thai alone from being consecutively generated, disrupting the feel of a dialogue. Through experimentation it was determined that after a maximum of three consecutive messages in either language, even if movement is still being detected on the respective side of the room, a message in the other language...
will be displayed, thus adding to the likelihood of the dialogue.

**Cinematic Effect**

Here resides the aesthetic core of the artwork, as image fragmentation and subsequent reconstruction processes define its visual identity.

Three very different creative minds and works inspired the algorithm behind the main animation: Marshall McLuhan, Ezra Pond and David Hockney. McLuhan (1988) claimed that the great technical possibilities of the cinematograph involved the “perceptions of simultaneities”. For him, this was achieved through a mosaic, as a world of intervals, not very dissimilar to what David Hockney suggests through his collages, criticizing the shortcomings of traditional photography, which limits the observer to a single, frozen perspective. He likens this frozen view to that of a paralyzed Cyclops (Vrobel 2011). Pond further unifies these two visions by stating that an image presents an intellectual and emotional complex in an instant of time. For him the image is not an idea, but rather a radiant node or cluster. He proceeds into calling it a *vortex*, “from which, and through which, and into which, ideas are constantly rushing” (Pond 1914).

“Hello” uses two different image sources: a set of 148 photographic images, categorized as either “people” or “landscapes”, and the webcam stream. Since “Hello” was developed in Processing, it takes advantage of the draw() function as a continuous loop through which different generations are produced. At any given moment one image is chosen as the main image. This main image is fragmented into a mosaic of sub-images, and each sub-image is modulated in terms of scale, rotation, tint and transparency:

\[
l_n = \sum_s s(x, y, f, c, \mu, \kappa_{-1}) \cdot r(x, y, f, c, \mu, r_{-1}) \cdot t(x, y, f, c, \mu, t_{-1}) \cdot t(x, y, f, c, \mu)
\]

Scale (s), rotation (r) and transparency (tr) modulation are complex procedures, combining trigonometry and controlled randomness as functions of spatial position, frame count, audience interaction (amount of detected movement) and previous values (previous generation). Tint (ti) is only modulated as a function of frame count and detected movement. The end result produces a sensation of flickering movement, as if the (still) images are in fact originally imbued with movement, as the mosaic evolves over time. The same modulation procedure is applied to an “interference” image, but only a small amount of its sub-images are displayed over the main image, as shown in figure 2. These interferences are subtle, yet quite visible when animated and they constitute an interpretation of both McLuhan’s perception of simultaneities and Pond’s vortex.

The consistent and subtle progression of a dominant hue and the use of transparency contribute to easing each new main image mosaic over its predecessor, thus providing the sense of a continuum, where new images take some time to fully develop and be perceived by the audience, documented by figure 3.

The webcam stream is used sparingly, very seldom retaining one of the captured frames (< 5%) for movement analysis and using it either as a main image (as depicted by figure 4) or an interference image. As they are interwoven in the animation flow, the audience will usually take some time before they fully process and perceive them, acting as an added element of surprise and bringing the public into the artwork itself.

![Figure 2 – The white circles show the interference sub-images over the main image.](image)

![Figure 3 – The merging of two consecutive main images as the dominant hue is shifting from green to yellow.](image)

![Figure 4 – A glimpse of the author, captured from the webcam stream.](image)
where trains are not only filled with Portuguese citizens, but also a very significant number of tourists, thus turning the soundtrack into a rather intricate mesh of different languages. These recordings were creatively mixed into four loops, corresponding to four floors of system excitement.

The excitement state (ES) is calculated into a variable as a function of the detected movement. This variable is controlled by a TTL mechanism based upon a weighted arithmetic mean, allowing for the soft transitioning between very different amounts of movement detection, fading in and out. The first loop corresponds to ES-0 (quiet room), is the default soundtrack and spans over two minutes and thirty seconds, looping throughout the whole exhibition.

The three remaining loops correspond to ES-1, ES-2 and ES-3, and were produced so that their playback can start and end seamlessly. They are consecutively triggered as the system reaches each of the three remaining excitement states. No new playback is started for a particular ES if the corresponding sound is already playing. The soundtrack thus mimics the events in the exhibition room, becoming very loud and frantic when ES-3 is reached, corresponding to a very agitated audience. As the movement quiets down so does the soundtrack, as each file playback (ES-1 to ES-3) reaches the end.

Hello was entirely developed in Processing 3.3.7. Its graphic vocabulary consists of image objects alone (PImage) and the soundtrack is delivered through overlapping looped samples manipulated with the Minim sound library.

These simultaneous events would prevent the author from being at both locations for the opening dates, thus the solution was to integrate interactive calibration into the artwork, taking into account the noise floor and movement detection threshold, including ES differentials. To this effect a small viewport depicting the ES can be rendered visible at all times, and direct testing under the actual exhibition room conditions allows to accordingly control the system sensitivity, as shown in figure 5.

Figure 5 – Calibrating system sensitivity with the help of a color-coded area of a small viewport, top center of the image.

Testing

Implications

The artwork was due for simultaneous exhibition in two venues separated by over 10,600 km. This posed another challenge since, as stated in the Movement Detection discussion, webcam noise needs to be taken into consideration in order to effectively distinguish between noise (mostly caused by poor lighting) and actual movement (caused by visitors), which means that the installation needs to be calibrated locally.

Figure 6 – Poster for the collective exhibition held in Faro and Bangkok.

Intervention

“Hello” was simultaneously presented at Clube Farense, in Faro, Portugal and at the Humble Projects gallery, in Bangkok, Thailand, as part of the “Conversa” (conversation) collective exhibition (figure 6).

The deployment at each venue was made according to the rider shown on figure 7, reproducing – to some extent – a cinema environment, while allowing for the free flow of visitors in the screening room.

Audience feedback

During the attendance of the Faro exhibition opening, the author was able to interact with several visitors, and assess their opinions and reactions through informal interviews. This choice of format is justified by its openness to incorporate different visions and insights, which may have been absent from the author’s intentions, thus allowing for a better exploration of audience perception.

Based on 15 informal interviews, the author was able to ascertain that:

- 5 Portuguese visitors correctly inferred that the first message in Thai was “Hello”, even though they had never seen the word before and admitted to being incapable of copying the writing. They developed some curiosity about the language and posed a number of questions about it;
- 8 visitors failed to establish a causal connection between their movements and the artwork’s behaviour, especially at times when there were too many people in the room. The correlation was more easily established when only one individual or smaller groups were present, because as soon as they walked into the room and their movement was detected, they were greeted by the “hello”
messages. This contributed to establishing a sense of cause and effect.

- 3 individuals, who came in a group, were intrigued by the dialogue flow and were discussing whether it was purely random (which it is) or if there was also a link to their actions. This added to their performance since they decided to explore different situations (sitting down, getting on their knees, jumping, etc.);

- a group of 5 visitors interacted the longest with the installation (3 minutes, over six times the mean amount of time spent, established by Smith, Smith and Tinio (2017)), and were actively collecting information from the dialogue and discussing it in order to understand the underlying plot. When asked about what triggered their curiosity they stated it was the message about “someone being robbed”;

- 1 individual stated that, when alone in the exhibition room, he purposefully placed himself out of the webcam’s range, thus avoiding the text-messaging aspects of the installation, just to appreciate the cinematic experience, which he described as “mesmerizing”.

- all 15 individuals confirmed the overall sensation of being in a train or in a train station and two of them even vented the possibility that the conversation (which was also the name of the exhibition) might actually be taking place between Faro and Bangkok in real time.

Conclusion

The present article is the first formal written intervention within the a/r/cographical project. It depicts and presents all seven stages of its creative development, from inspiration to intervention, as well as the findings that were gathered during the attendance of the Faro exhibition opening.

The relationship the artwork established with its interactors managed to create both an individual and group performative and interpretive experience. However, smaller group interactions seemed to be the most successful in establishing the interaction correlations, particularly in collaborative exploration of the text-messages and finding meaning in the artwork.

As the curiosity to unravel the underlying story increased, so did the audience restlessness, often disregarding any initial constraints or awkwardness related to self-awareness. Speculation and interaction with other audience members then became almost natural, in a collaborative effort to decipher the plot. This suggests that some of the utterances, in future versions, may become more impactful (quizical, strange or even shocking), in order to better capture the audience’s attention.

Departing from very incomplete information some audience members were able to identify the key aspects of the underlying narrative, based upon key utterances, and regardless of their order.

The narrative was thus built fragment by fragment, as Ezra Pond suggested in his ideogrammic method, the culmination of the periplum, as the method of “presenting one facet and then another until at some point one gets off the dead and desensitized surface of the reader’s [audience’s] mind, onto a part that will register” (Pond 1970).

These conclusions are in line with Dewey’s statement (1980) that “the product of art – temple, painting, statue, poem – is not the work of art. The work takes place when a human being cooperates with the product so that the outcome is an experience that is enjoyed because of its liberating and ordered properties”.

Acknowledgements

This artwork was developed within the “Cultural Adventures” research project, hosted by the University of Technology of Architecture and Design of King Mongkut Thonburi, Thailand, and the Research Center for Arts and Communication (CIAC) at the University of Algarve, Portugal. Conversations with Phanuphong Songkhong played an important role in the early stages of this project. The author would also like to acknowledge the significant contributions from students at the King Mongkut’s University of Technology Thonburi, who played a crucial role in providing the Thai utterances, as well as their translations.
References


Which type is your type?

Jéssica Parente\textsuperscript{1} and Tiago Martins\textsuperscript{1} and João Bicker\textsuperscript{1} and Penousal Machado\textsuperscript{1}

\textsuperscript{1}University of Coimbra, CISUC, DEI

\{jparente, tiagofm, bicker, machado\}@dei.uc.pt

Abstract

An interactive evolutionary system to generate letterings is presented. The system allows the creation of a wide range of alternative designs that can be used as stimuli for inspiration or for the creation of visual identities with different variations. This work began as a parametric system that generated glyph designs by recombining parts of skeletons extracted from existing typefaces which are then filled with custom shapes. In this paper, we employ a Genetic Algorithm to evolve the input parameters of this parametric system. The experimental results show that the presented evolutionary system enables users to interactively create unique letterings according to their aesthetic preferences.

Introduction

As human beings, the way we communicate is one of the most unique characteristics that define us. According to Cheng (2006), typography is the visual manifestation of language and the instrument that turn characters into words and words into messages. In the modern world, typography is also a way of “given meaning”, it needs to transmit resonance and depth to the messages it is transmitting. To communicate a message, a designer can use the composition of the elements and typography. Therefore, the design of type can be useful to add layers of meaning (Cheng 2006; Shaughnessy and Bierut 2009).

With the technological revolution, type design tools have changed. Now, typefaces have to be optimised taking into account where they’ll be read and the target audience. However, as Adrian Schaugnessy (Shaughnessy and Bierut 2009) points out, today’s type designers continue to do what they have always done: they are changing and adapting to developments in technology, media and literacy. The technology made possible new ways of exploration and allowed the type designer to explore previously unthinkable fields. Consequently, through these new possibilities, more typefaces emerged, but also the uncertainty of their quality. Moreover, with the emergence of Artificial Intelligence (AI), in the twentieth century, the potential of machines to be creative in their way can now be explored by academicians and practitioners from diverse disciplines. In the typography field, appeared useful tools that provide a wide variety of alternative designs that promote new ideas during the design process. These new tools offer good support in the design process which can be an advantage when compared with conventional tools. By taking advantage of these computation systems we can find inspiration and unlock a creative block. However, these systems should respect some typographic rules by creating a balance between what the user can change and what the system automatically performs.

With that in mind, we decided to create an Interactive Evolutionary Computing (IEC) system that generates letterforms, providing a wide range of alternative designs that can be used as stimuli for inspiration or even for the development of visual identities with different variations (Lupton 2006; Shaughnessy and Bierut 2009; Lehni 2011).

This project began with the creation of a less sophisticated system that designed glyphs by the combination of skeletons of existing typefaces and posterior filling. To develop the system three aspects were worked out: (i) the development of the structure of the typefaces generated and the codification of the different elements of the structure of the letter in different layers; (ii) the combination of layers of different typefaces; and (iii) the creation of glyphs through the generation/ modification of the elements of these layers. The IEC system uses the drawing process of the parametric system and evolves the glyph design parameters. An example of a generated lettering can be found on Figure 1. Each population is composed of letterings, so each individual represents a sequence of glyphs. We implement a Genetic Algorithm (GA) to evolve a set of parameters that control the generation of letterings. Then, we created a graphic user interface to allow the interactive guidance of the evolution process.

The remainder of this paper is organised as follows. Related Work Section presents related design projects in the domains of type design and IEC systems. Approach Section describes the Parametric and the Evolutionary System. Experimentation Section validates and demonstrates the potential of our system as a computer-aided creativity tool and discusses the achieved results. Finally, the Conclusion and Future Work section summarises our work and presents future research directions.

Related Work

In the early 1990s, the evolution of software to design fonts opened doors to new methods to create type. The Beowulf font (FontFont nd) appeared alongside with the first series
of fonts with random outlines and programmed behaviour. They make use of a kind of pre-programmed randomness, they randomly move the points of the contours to deform the glyphs and, due to this there are not two identical glyphs. Today it is also possible to create glyphs that change according to data. Typography Music (Silanteva 2011) is a system that generates glyphs that react to music. The glyphs are formed from a grid and constructed by the combination of layers. Each layer is constituted by a range of modules and the shape of each module changes with the type of music, for instance for an organic sound the modules are circles, for an analogue sound, they are octagons.

With these new possibilities, computational creativity becomes a topic of discussion and computational systems that try to imitate human creativity appear. In 1992, Gary McGraw and Douglas Hofstadter proposed (Rehling and Hofstadter 2004), a system for the automatic generation of the lowercase letters of the roman alphabet in different but internally coherent styles. Starting with one or more seed grid-based letters, the system attempts to create the rest of the alphabet in such a way that all letters share that same style. Nowadays, we also see an increasing attempt to humanised computational intelligence. Interactive Evolutionary Computation (IEC) is one of these research domains and it is used in diverse research categories, for instance, design and computer-generated animation, music, face image generation, speech processing, image processing, among others. IEC as an optimisation method that involves Evolutionary Computation (EC) and it optimises a target system based on a user’s subjective evaluations. EC systems had the advantage of providing a wide range of alternative designs that can be used as stimuli for inspiration. Besides, when we use an IEC system, we can blend the capabilities of EC optimisation with human evaluation and make fuller use of both of them. An experimental evaluation involving two types of interaction styles — Direct Manipulation and Interactive Evolutionary Design — to do creative tasks in a type design system can be found in (Lund 2000). The goal of this research was to compare these two kinds of interactions and it was found that that direct manipulation prototype offers a higher degree of freedom to design typefaces. Direct Manipulation is more suitable when the objective is clearly defined. On the other hand, Interactive Evolutionary Design proved to be the interaction where the users were more active; and it is more suited for creative tasks. Good solutions can be explored with the use of IEC systems; they can adapt, select and create even “better” solutions from a generation to another. With that in mind, Jaksa Kuzma and Sincak created a system (Kuzma, Jakša, and Sincak 2008) that helps the user to create typefaces. The user’s evaluation affects the evolving process. The structure of the typefaces is based on Computer Modern font and it has several parameters to change the design of the glyphs. Alphabet Synthesizer Machine (Levin, Feinberg, and Curtis 2001) is another system that has the same goal, the system creates abstract alphabets from a writing simulation using a Genetic algorithm (GA) — a search heuristic that is inspired by Charles Darwin’s theory of natural evolution. The developed algorithm evolves a population of candidate glyphs according to a set of fitness metrics established by the user. The developed glyphs evolve as individuals to improve their characteristics, and as a species. Genotyp (Schmitz 2004) is another similar system, it generates typefaces by combining genetic characteristics of different fonts. The system allows the combination of different fonts and manipulation of their genomes. The combination of different fonts can result in mutations, but the granted inheritance can be modified later manually (Schmitz 2004). Evotype (Martins et al. 2015; 2016; 2018) is a project, divided into iterations, that explores different ways of designing glyphs. In the first iteration, the glyphs were designed by the combination of line segments arranged in a rectangular grid. Then, they were evaluated according to the visual similarities they had concerning the previously selected font (Martins et al. 2015). In the second iteration, glyphs were designed by the combination of shapes inserted by the user. To the evaluation part, they used the evaluated system of the last iteration combined by a classification model trained with numerous existing typefaces (Martins et al. 2016). In the last iteration, a stencil approach was created in which they generate stencils com-
posed of line segments. This iteration allows the design of all letters in a more coherent way (Martins et al. 2016; 2018). Some approaches explore EC systems of type design creation outside the Roman alphabet. (Fischer 2004) is a tool that supports genetic operators by the addiction of variations to the design of the fonts based on Bézier curves. (Unemi and Soda 2003) is another system for font design that uses IEC technique, the system creates Japanese Katakana from very simple stroke elements. To create the genome of each individual they encoded some parameters for drawing elements. The initial population has sixteen individuals with random genes and the user can breed the font that he/she liked most.

**Approach**

In this paper, we present an IEC system that generates letterings. Our goal for the system was to create something that could be useful for designers to develop glyphs and letterings. The first part of this section presents the initial stage of this project, a parametric system to generate typefaces. The second part describes an evolutionary system which evolves the parameters of the previous glyph design system. We also created a graphic user interface to help the user to control the IEC system. We present the system, its possibilities and, in the end, we validate it.

**Parametric System**

This project began with the creation of a parametric system to generate typefaces. The system creates glyphs by the extraction of the skeleton of existing typefaces and separation of the skeletons into parts. The skeleton of each generated glyph will be the result of the combination of parts of the skeletons of different typefaces. Then, the generated skeleton is filled by the drawing of shapes repeatedly all over the skeleton. For more information, you can consult the first iteration of the parametric system on (Parente, Martins, and Bicker 2018).

**Developing the structure of the glyphs** For us, the structure of the generated glyphs was a import part to take into account. Nowadays, several typographic and generative systems develop typefaces, but most of them mostly focus on the letters’ filling and use, for the structure, hand-drawing typefaces which are mostly static. Our first major goal for this project was the creation of the structure of the glyphs. For that, we decided to use existing typefaces and extract their skeletons. Therefore, we make sure that the design of the glyphs follows the rules of traditional font design.

There are diverse works that explore skeletons extraction through the use of different methods these days (Naccache and Shinghal 1984; Gonczarowski 1998; Dimauro, Impedovo, and Pirlo 2011). We decided to use the Zhang-Suen Thinning Algorithm (Zhang and Suen 1984) that aims to extract the structural lines of a binary image. This algorithm receives an image consisting only of pixels of two different colours, for example, black and white, and returns it modified. All the pixels that are not essential to understand the image are removed, which is ideal for this work, where we need to extract the skeleton of typefaces.

Once the possible solution for the skeleton extraction was found, we decided to implement it in our project. The skeleton extraction process begins with the scanning of all pixels. If the evaluated pixel fits a series of conditions is defined as white. The process repeats over time and ends when none of the pixels suffers any alteration.

**Division of the skeleton into strokes** With the skeletons extracted, we needed to find a method that identified the different parts. After analysing the generated skeletons, we noticed that when a point was part of three segments it divides different strokes of the skeleton. A stroke is composed by at least two points, but in general, they are more, and a set of strokes compose a skeleton. To test this theory, we scanned all the points of the skeleton and when we found a border point, we created a new stroke, and so on till the end. Figure 2 presents the process of skeleton extraction and division into parts of a glyph for the letter “h”. The circles highlight the border points and a different colour was applied to differentiate the different parts detected.

**Skeletons recombination** One of our goals was to combine different typefaces’ structure into a glyph. Since we already had a system that generated skeletons, the next step was the combination of parts between different skeletons. To each glyph we needed to assess which stroke of each skeleton could be associated with each other. To do that, we determined the angular velocity and the central point. Therefore, each stroke did not need to be equal to the other to be pairing. To put it in context, angular velocity is a vector that represents the process of changing the orientation of a given line. For a line, the value will be equal to 0 and it will increase as it becomes more curved. The central point is an average of all points of the part in question.

The pairing process started with the ordering of the strokes of the first skeleton, from the longest to the shortest length. In principle, the error that could arise from the combination would be minimised. Then, each of these parts was compared with all the constituent parts of the second skeleton and so on. When the compared parts had a similar angular velocity and centre point, they were considered corresponding and we moved on to the next skeleton. At the end of the cycle, we had, for a given character, several versions for each stroke.

**Give body to the skeletons** In the section Developing the structure of the glyphs we mentioned that the pairing process started with the exclusion of the pixels furthest from the centre of the stroke. Therefore, when calculating the distance between each pixel of the skeleton to the nearest pixel from the border, we determine the width of the original typeface. With this measure, we can replicate the glyph.
or increase or decrease the weight proportionally. Then, using different shapes (e.g. circles, triangles, squares, or other abstract shapes) repeatedly we fill the strokes of the final skeletons and generate typefaces using different colours and transparencies. The filling of each stroke is composed of modules repeated along the stroke line. To each generated typeface, we could determine which typefaces we wanted as input and we can choose the colours and modules to use.

**Evolutionary System**

Although the parametric system was already capable of generating typefaces, we wanted to explore the system more. Each generated glyph had a series of parameters that still had a lot to explore. Besides, parameters such as the density of shapes repeated in the filling were not even used as a variable. Thus, a system to evolve the input parameters of the parametric system can provide us with new ideas for the creation of the glyphs by the generation of unpredictable designs that could be something useful as stimuli for inspiration. With that in mind, we decided to employ an IEC system that generates letterings. To achieve this, a GA is implemented to evolve different populations of letterings which are based on the designing process of the parametric system developed earlier.

**Representation** The evolutionary system evolves a population of letterings. Therefore, each individual represents a sequence of glyphs. The genotype of each individual consists of a list of tuples containing integers. Each tuple represents a stroke of a glyph. Each integer encodes the index for a given attribute for the stroke that is represented by the enclosing tuple. The attributes of each stroke (see Figure 3) are the input typeface, the used shape, the shape’s scale, the number of shapes or density, the shape’s colour, the opacity of shape’s fill, the opacity of shape’s contour, the skeleton’s opacity and the width of the shape’s contour. The filling of each stroke is composed of the repetition of shapes along the stroke line. The phenotype of each individual is the lettering generated with the parametric system already described using the settings encoded in the genotype.

**Crossover, mutation and evaluation** The generated glyphs are the result of the user’s subjective evaluations. During the evolutionary process, the users choose their preferred letterings from the current population. This leads to the creation of more populations containing individuals with visual properties that go according to their taste. To do so, we use the individuals selected by the users and then we apply genetic variation operators, including crossover and mutation. The crossover operator breeds the selected individuals by recombining their genetic information. Then, the mutation operator permits the variation of genes in the genotype.

When the user selects multiple individuals, we apply the crossover operator to random pairs of these individuals and then apply the mutation operator to the resulting offspring. When the user selects only one individual, we use the mutation operator to create variations of it. When the user selects no individuals, we apply this procedure to the last selected individuals, which may not necessarily be of the latest generation.

**Visualisation** We developed a graphic user interface to enable the user to visualise, select and breed letterings (Figure 4). The individuals (letterings) of the current generation are arranged on a grid to facilitate users’ choice. The evolutionary process begins with the selection of the values to each parameter by the users. More specifically, they can choose the possible values for the skeleton, shape, scale, density, colour, fill-opacity, contour opacity, skeleton opacity and contour width. Additionally, users can also write the letters that will compose the evolving lettering. During the evolutionary process, users can interactively pick the lettering(s) that they liked the most. Also, at any moment of the evolutionary process, users can export the evolved letterings as vector files to, for example, apply them in a design process or make further design refinements.

<table>
<thead>
<tr>
<th>Input skeleton/</th>
<th>Shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>typeface</td>
<td>▲</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Shape's scale</th>
<th>Number of shapes/density</th>
</tr>
</thead>
<tbody>
<tr>
<td>□</td>
<td>□</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Shape's colour</th>
</tr>
</thead>
<tbody>
<tr>
<td>▼</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Opacity of the shape's fill</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Opacity of shape's contour</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Skeleton's opacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Width of the shape's contour</th>
</tr>
</thead>
<tbody>
<tr>
<td>□</td>
</tr>
</tbody>
</table>

![Figure 3: The attributes of each glyph stroke. The visual elements marked with * are not visible due to their low opacity.](image-url)
Experimentation

As previously mentioned, our IEC system allows the generation of very diverse letterings. This leads to numerous different possibilities to represent the same word. This occurs thanks to the big range of parameters that the system uses to create each glyph. In this section, we summarise the visual possibilities of this approach, wherein different parameter settings and analysed the results. During these experiments, we studied the impact of the parameter settings.

The values of each parameter are the following: (i) to the input skeletons we use eight typefaces as input (Didot, Times New Roman, Adobe Caslon Pro, Bodoni, Helvetica, Adobe Garamond Pro, Futura, Baskerville); (ii) to the shape parameter we decide to design 16 shapes; (iii) to the shape’s scale, we established 4 values (from half to twice the size of the input typeface’s width); (iv) to the density parameter, we established 6 proportional values to the font-size of the glyph (That way when the font-size is smaller the generated glyph had a lower level of detail); (v) we established ten possible colours to the shapes; (vi) to the opacity of fill, contour and skeleton we set 6 values (from 0 % to 100 % of opacity); and (vii) for the contour width we decided to use just 5 values. In the beginning, we established four possible values to the opacity of fill, contour and skeleton however, the probability of having a blank shape was too high. The users can determine which available values of each parameter to use. After some tests, we decided to keep the population size of 15 individuals and the mutation rate of 0.1.

According to the users’ desire, diverse letterings can emerge by the alteration of the values for each parameter and posterior evolution of populations through the selection of favourite individuals. To understand the possibilities of the parametric system, we decided to explore the available parameters. Then, we ask two different users to evolved letterings are to validate the IEC system.

Generating Letterings

We began the exploration of our parametric system by the attempt to create glyphs similar to the common typefaces. We decided to use the word “maria” and limit the parameters to generate letterings with circles as the shape, black as the only colour and with 100% of opacity. Bodoni was the typeface chosen as the input to generate the skeleton and, to the scale, we use the scale of the original typeface. Figure 1 — Variation 1 shows the result. In the second variation of Figure 5, we let the system generate letterings with the same characteristics as the previous, but with more typefaces as inputs, typefaces that were similar to the one used above. In Variation 3 of Figure 5 we add more typefaces but more different than those used previously (Variation 4 of Figure 5) we use two very different skeletons, we use Bodoni and Helvetica typeface. By the observation of the letterings generated Figure 5, we can notice that the system can combine more than two typefaces and generate a third one by mixing their skeleton. We also see that the parts of each glyph are unique, even with the same parameters the system can generate variants.

Figure 5: Original glyphs (top) and variations (bottom) created by recombining strokes of the original glyphs with strokes of glyphs of other typefaces.

We also explored the variations on the shape’s scale (Figure 6). We wanted to see the different behaviours of the system and their adaptation to different values of the shape’s scale. We used just one typeface as input and we use the same black circles used previously and we employ opacities of 0 and 100% in the fill of the circles, Figure 6 and Figure 7 respectively. We noticed that the letterings generated in Figure 6 had more contrast, and for that reason, the differences between strokes became a lot more visible. Becomes interesting to observe the differences that the same lettering suffers only by changing the scale of the shapes. Besides, these variations in black and white with the opacity of the shape of 100% (Figure 6) are the ones that can be used more easily in a text, mostly the glyphs composed by strokes that have a width equal or less the original typeface used as input.
From our point of view, the contrast on the glyphs’ skeleton present mostly on Variation 3 and 4 of Figure 6 can be very useful in the design context, use as data-driven logotypes or dynamic identities. On the other hand, Figure 7 presents a series of variations that had more detail, and that could be, more easily used to create an identity. Besides, the shades created by the overlap of the layers create another variant in the lettering.

Figure 6: Four variations in which we vary the shape’s scale while maintaining the opacity of its fill set to 100%.

Figure 7: Four variations in which we vary the shape’s scale while maintaining the opacity of its fill set to 0%.

In this system, we can also work with colours, and use different kinds of shapes in the design of a glyph. In Figure 8 and 9, we let the system generates letterings varying three parameters. In lasts variations, we decide to have a bigger level of abstraction in both Figure 8 and Figure 9. Each variation uses different values of scale, similar to the previous examples. The use of colours adds a new level to explore, besides the use of different opacities serves to mix different parts of the glyphs. The use of shapes without fill (Figure 9) adds much more levels of details, and it allows us to see all the shapes that compose the glyph. Thanks to the large variety of generated letterings they could be part of a composed of a dynamic identity with many variations.

Figure 8: Four variations in which we vary the shape, its colour and scale while maintaining the opacity of its fill set to 100%.

Figure 9: Four variations in which we vary three parameters. In lasts variations, we decide to have a bigger level of abstraction in both Figure 8 and Figure 9. Each variation uses different values of scale, similar to the previous examples. The use of colours adds a new level to explore, besides the use of different opacities serves to mix different parts of the glyphs. The use of shapes without fill (Figure 9) adds much more levels of details, and it allows us to see all the shapes that compose the glyph. Thanks to the large variety of generated letterings they could be part of a composed of a dynamic identity with many variations.

Evolving Letterings

In the last section we presented some of the visual possibilities of the parametric system. Now we pretend to demonstrate the potential of our IEC system applied to a design point of view. An analysis of the evolution of fitness across generations would be mostly pointless since we are not interested in demonstrating that genetic algorithms work (that has been established countless times before). Instead, we are interested in validating and demonstrating the potential of our system as a computer-aided creativity tool. For that purpose, we focus on the analysis of the results obtained by different users when working with the tool (Figure 10 and Figure 11).

The user A (Figure 10) decided to use the word “create”. He uses all available shapes, typefaces and values of scale and density, but only four colours. The user ended up choosing letterings where the glyphs could be read perfectly. However, he believed that most of the generated individuals, even those never chosen, could be a possibility for application in an identity. Throughout the generations, several valid hypotheses were assumed by the user, but he wanted to test more to see where the system could take him. That is one of the advantages of this system. It is capable of generating non-expected versions that can unblock an artistic block.
The user B (Figure 11) decided to use the word “Anna”. She used just part of the available shapes and colours and she decided to evolve letterings with skeletons composed with shapes without fill. Through the generations, the results were diverse, but the system ended up converging for the style she wanted, something more light and clean.

The diversity of the results highlights the expressive power of the tool and the impact of the user’s preferences on the outcomes of the system. It is totally possible that if the same users use the tool one more time with the same established parameters, they would create different generations. Another interesting thing about the evolutionary system is that it can create non-expected combinations. Nowadays, as designers we need tools to help us in the creative process, not to replace us. We need a combination between what the system generates to please us, according to our choices, and what the system does to create something more.

**Conclusion and Future Work**

This paper presents an IEC system that generates letterings. The system is composed by a parametric system that extracts the skeleton of existing typefaces, separate the skeletons into parts, recombine the parts in a final skeleton and fill the final glyph by the drawing of shapes repeatedly all over the skeleton. The second part of the system is an evolutionary algorithm that evolves parameters of the design of the glyphs. We demonstrate the system and its possibilities and validate them. Our main contributions include: (i) a parametric system capable of automatically create letterings; (ii) a evolutionary system that evolves the drawing parameters of the glyphs; (iii) exploration of different parameters of the design of a glyph into visual components; and (iv) an investigation into how evolutionary computation can be used in the field of type design.

It is important to highlight that it is a work in progress and for that reason, there is a set of things that we would like to do in the next iterations. In the actual system, the glyph that composed the letterings (individuals) are separated from each other. That means, that the glyphs of each word have no connection between them. In future work could be interesting to have some points in common in the glyphs of the same individual, at least in the glyphs corresponding to the same character. We also want to explore more the evolutionary by adding a system to save letterings that the user likes more. It would also be interesting that the system saves all the choices of the user. At this time, the system only looks for the lasts choices of the user. By making this alteration the system should adapt better to each user. In the future, we intend to conduct user studies in order to evaluate the evolutionary capabilities of the system, namely its ability to produce different and novel types that satisfy users’ preferences, as well as the system’s ability to promote users’ creativity leading them to explore new ideas and concepts.

**References**

Figure 11: Generation 1 (top) and 8 (bottom) of the evolutionary process guided by the user B. The selected letterings of the generation 8 are the ones that the user B liked the most.

International Journal of Pattern Recognition and Artificial Intelligence 07.


Acknowledgments

This work is partially funded by national funds through the FCT - Foundation for Science and Technology, I.P., within the scope of the project CISUC - UID/CEC/00326/2020 and by European Social Fund, through the Regional Operational Program Centro 2020, and under the grant [SFRH/BD/148706/2019]
Predicting A Creator’s Preferences In, and From, Interactive Generative Art

Devi Parikh
Georgia Tech & Facebook AI Research
parikh@gatech.edu

Abstract
As a lay user creates an art piece using an interactive generative art tool, what, if anything, do the choices they make tell us about them and their preferences? Both within the generative art form, and otherwise? As a preliminary study, we collect preferences from 311 subjects, in a specific generative art form and in other walks of life. We train machine learning models to predict a subset of preferences from the rest. We find that preferences in the generative art form cannot predict preferences in other walks of life better than chance (and vice versa). However, preferences within the generative art form are reliably predictive of each other.

Introduction
Generative art is art that has at least some of its features determined by a non-human autonomous system. The autonomous system is typically a computer, and it frequently relies on randomization to determine the art features. In the example in Figure 1 for instance, the start and end points of each curve are selected randomly. Among the human-defined features, some might be fixed by a generative artist (e.g., the fact that the piece is formed by a sequence of random points connected via Bezier curves), and other parameters might be open to manipulation (e.g., thickness of the strokes, color palette). Different settings of these open parameters, combined with the machine-defined features, leads to different instances of the generative art form. Generative artists often make interactive tools available so a lay person can set values of these open parameters and create their own generative art piece. We refer to these tools as interactive generative art tools, and they are the subject of this study.

We ask the question: what does the choices a lay person makes while creating art using an interactive generative art tool tell us about them – about their personality or preferences in food, fashion, interior design, etc., as well as about their preferences in the specific generative art form?

Effectively predicting their preferences in the specific generative art form can lead to a smarter interactive generative art tool. It can help the user create an art piece they like faster by encouraging them to explore a certain part of the parameter space. It can prevent the user from losing interest by discouraging them to explore a different part of the parameter space. Predicting their preferences in other aspects of life can position interactive art generation as a generic personality assessment tool for products and experiences recommendations. Finally, predicting their preferences in generative art from other known preferences in life can lead to improved art recommendation.

As a preliminary study, we conducted a survey where 311 subjects consented to participate and self-reported their preferences along various parameters in a generative art form (Strokes, Figure 1), as well as in various walks of life such as food, chocolate, alcoholic beverages, music, interior design, fashion, paintings and their other traits such as gender, personality type, exposure to design principles, artistic inclination, and introspectiveness. We train machine learning models to predict subsets of these preferences from other preferences. We find that user’s preferences in other aspects of life cannot be reliably predicted from their preferences in the generative art form (and vice versa). However, their preferences in the generative art form can be predicted with statistical significance from other preferences. We find that user’s preferences in other aspects of life cannot be reliably predicted from their preferences in the generative art form (and vice versa). However, their preferences in the generative art form can be predicted with statistical significance from other preferences. We find that user’s preferences in other aspects of life cannot be reliably predicted from their preferences in the generative art form (and vice versa). However, their preferences in the generative art form can be predicted with statistical significance from other preferences.

Related Work
Preferences in art and personality. (Özpolat, and Taşkesen 2015) study the correlations between five personality traits and art preferences among 24 visuals from Renaissance, Cubism, Abstract Art, Traditional Art, Impressionism and Surrealism. (Chamorro-Premuzic et al. 2008)
Figure 2: Configurations of Strokes we studied and % times each alternative (b-i) was preferred by subjects over the default (a).

performed a similar study across over 90,000 individuals in the UK. (Chamorro-Premuzic et al. 2010) find that personality traits correlate better with art categories when these categories are defined based on emotional valence and complexity as assessed by a collection of observers, than categories defined by researchers or historical art taxonomies. (Ercegovac, Dobrota, and Kuščević 2015) also study the relationship between personality traits and art preferences, but for both visual art and music. They also investigate correlations between music and visual art preferences. (Gridley 2013) study correlations of visual art preferences with personality traits as well as styles of thinking. They also find evidence for cross-modal relations in aesthetic preferences across food, music, and visual stimuli. (Lyssenko, Redies, and Hayn-Leichsenring 2016) study how participants describe abstract artwork and the relationship of these descriptions to various image properties. They also investigate the correlation between personality traits and preferences. To the best of our knowledge, connections between art perception and preferences has not been studied in the context of generative art (the focus of this work) or even digital art in general. As the landscape of art changes with the incorporation of digital tools and more recently AI, it is valuable to consider how these tools can be made smarter to better assist a human in their creative process. The ability to predict a user’s preferences is central to such smart tools.

(Bhattachariya 2016) discusses models of preferences in the context of computational creativity to embrace the subjective nature of evaluating creative value. (Cook and Colton 2015) design a software system capable of having preferences – to make and justify subjective decisions beyond using random chance or a pre-defined external heuristic.

Casual creators. The interactive generative art tools we study fall in the category of “Casual Creators”. This is in contrast with tools that are designed to assist professionals or amateurs in their creative process towards a specific goal. Casual creators on the other hand are autotelic creativity tools that cater to enjoyable explorative creativity over task completion. (Compton and Mateas 2015), who coined the term, stress the interactive aspect of casual creators where the user is the driver, and the creating process as being core to the experience. A casual creator is an effective tool if it helps users find desirable artifacts without getting stuck in a local minima or being lost in a vast space of bad artifacts. Our work addresses exactly this. The various parameters in the interactive generative art tool define the space of artifacts, of possibilities, that a user can explore. Our work predicts the user’s preferences. A computational model that uses these predictions can influence the path the user takes in this space; it affects the probability that a user will encounter a certain artifact in their creating process. By effectively predicting their preferences, we increase the chances that users will find a desirable artifact when using a casual creator.

Individual user preferences. Existing work, e.g., (Zsolnai-Fehér, Wonka, and Wimmer 2018), models individual preferences of a user as they explore a parameterized design space. Our work learns correlations between preferences across parameters from a population of users. The two directions have a common goal – helping a user find designs they like – but are complementary. Game content generation has been personalized for both designers (Liapis, Yannakakis, and Togelius 2013) and players (Shaker, Yannakakis, and Togelius 2010).

Interactive Generative Art: Strokes

We design our study around Strokes (Figures 1 and 2) as the generative art form. We chose Strokes because it is an abstract form, allowing us to focus our study on visual preferences rather than semantic associations.

A Strokes piece is a series of overlaid shapes. A shape is started by connecting two random points on a square canvas via a curve of a certain thickness ($T$). The curve may be a straight line, or a quadratic Bezier curve using a third point as a control point. This control point is the midpoint between the two end points perturbed by random noise. The noise is uniformly random in the range $R$, which is 10% to 20% the width of the canvas. This noise is either added or subtracted to the $x$ and $y$ co-ordinates of the mid-point. Each of these 4 possibilities has a probability $\nu$ of 0.25.

Having placed the first curve, the end point of the curve is connected to another random point on the canvas via a curve. This process is repeated. After each curve is drawn, the shape either continues (with probability $P = 0.5$) or the shape ends and a new shape begins. When the shape ends, the canvas enclosed by the curves and a straight line connecting the start point of the first curve in the shape and end point of the last curve in the shape is colored by a random color from a palette. The color of the curves themselves is a pre-defined background color in the palette. When a total of $N$ curves have been drawn, the last shape ends and the piece is complete. Any pixel covered by more than one shape is colored by the most recent color.
The generative artist designed this generative process, chose the colors in each palette, the probability $P$ with which a new shape starts after each curve, the range $R$ that determines the amount of noise added to a mid point, and the probability $\nu$ of adding noise to each of the 4 directions to form the control point of the quadratic Bezier curve (when applicable). The machine picks the random end points and noise added to the mid point to form the control point of the quadratic Bezier curve (when applicable). The color palette, number of curves $N$ in the piece, thickness of curves $T$ and whether the curves should be a straight line or a quadratic Bezier curve are free parameters. These parameters are provided as options on an interactive tool as seen in Figure 1. The random seed is kept fixed when a user is changing parameters on the interface so that the only influence changing the piece is input from the user. The user can click on the “Generate” button to change the random seed that determines the machine’s influence. A video demonstrating the interface is provided here https://youtu.be/YzFzjkKNNMo.

As options, the tool provides 6 color palettes, 11 densities which determine the number of curves ($N = 2^{\text{density}}$), 15 line thicknesses, and a binary option of curved or straight lines. In our study, we restrict the number of palettes to 3, densities to 3, line thicknesses to 4, and retain the straight vs. curved lines option. We start with a “default” configuration for each of these options and generate 8 versions of the piece by changing one property at a time (Figure 2).

**Collecting Preferences**

We collected 36 preferences – 24 in Strokes interactive generative art, 12 in other walks of life – from 311 subjects on Amazon Mechanical Turk. Subjects were from the US, had completed 5000 tasks on AMT with an approval rating of ≥95%, and were paid higher than minimum wage in the US.

For Strokes, we generate 8 pairs of comparisons: default in Figure 2a vs. each of the 8 edited versions in Figure 2b-i. Both pieces in each pair are generated with the same random seed so that there is only one cause of variation between the two pieces, but different seeds are used across pairs to ensure that the preferences we collect are generic across seeds. We randomly order the default and the edited version within a pair. The 8 pairs are also randomly ordered. We generate a total of 3 sets of these comparisons with different random seeds. This gives us 24 two-way forced choice pairs in the context of interactive generative art. For each pair, subjects were asked “Which visual pattern appeals to you more?” Including options for no or equal preference may be better.

For other walks of life, we ask subjects for 12 preferences. (1) Do you reflect on a regular basis (e.g., write in a journal)? Yes/No (2) Which do you prefer? Milk vs. dark chocolate (3) Which do you prefer? Wine vs. beer (4) Which do you prefer? Country vs. rock music (5) Do you have any exposure to design principles? Yes/No (6) Are you artistically inclined? Yes/No (7) Which gender do you associate with more? Male/female (8) What personality type do you associate with more? Introvert/extervert (9) Which do you prefer? Sweet vs. savory food (10) Which of these styles of painting appeals to you more? Cubism vs. Renaissance (examples were shown) (11) If you could set up your home however you liked, which of these styles would you go with? Modern vs. traditional (with examples) (12) Irrespective of your gender, which of these fashion styles do you relate to more? Bohemian Chic vs. business casual (with examples).

**Are people self-consistent in their preferences in generative art?** Recall that 311 subjects were shown 8 generative art comparisons for 3 seeds. Across these 2488 sets of 3 responses, we check how accurately a response to a 3rd seed can be predicted by assuming the same response to the other 2 seeds. The prediction accuracy is 81%. When the response to the 2 seeds is different, we broke ties using the prior. Recall that each pair contains the default configuration and one of the eight alternative versions (Figure 2). Across subjects, the default configuration is preferred 62% of the time, and was used to break ties. Note that always predicting that subjects prefer the default option would result in a prediction accuracy of 62% – significantly lower than 81% reported above. Overall, it is clear that subjects frequently prefer alternative configurations and they are consistent in this preference across seeds. Thus, there is scope for predicting the personal preferences of a user automatically. Figure 2 shows the % of times each alternative is preferred over the default.

**Which preferences are related?** We omit detailed statistics for space considerations. We find that gender is the best predictor for wine vs. beer preference. Preferring gray palette over the default is the best predictor for preferring straight lines over the curved lines in Strokes. Preference for the bright palette is a good indicator of preferring thicker lines. Unsurprisingly, preference for the sparser pattern is the best predictor for preference for the sparest pattern. Overall, we see promising correlations across preferences.

**Predicting Preferences**

To predict preferences from other preferences, we train a variety of ML models. We create three groups of preferences: $A$: 8 art (Strokes), $L$: 12 other walks of life, $U$: all 20. Each group can be used as features $F$ to predict preferences in the other group (target $T$). A group can also be used as features excluding that target preference from the input features. This results in a total of 9 settings $(F,T) \in \{A,L,U\} \times \{A,L,U\}$ where $\times$ denotes the cartesian product. We train and test our models via leave-one-out cross validation; train on preferences from 310 of 311 subjects, test on the remaining subject, repeated 311 times. To normalize for different priors for different preferences, we report class normalized accuracies. We experiment with: nearest neighbor, logistic regression, linear Support Vector Machines (SVMs), polynomial SVMs, Radial Basis Function (RBF) SVMs, neural networks, decision trees, and a matrix completion approach. Models for learning from preference data specifically may be better.

Linear SVMs perform the best. Models are unable to predict interactive generative art (Strokes) preferences based on other life preferences (50.00%) and vice versa (48.80%). Chance performance is 50%. However Strokes preferences predict other Strokes preferences well (64.33%). Knowing other preferences in life further helps predict Strokes preferences (65.11%). We focus on these two settings: using
Strokes preferences to predict other Strokes preferences, and using all preferences (Strokes + other walks of life) to predict a held out Strokes preference. Nearest neighbor is an informative point of comparison. It assumes that a test subject’s preference is the same as the training subject who has the most other preferences is common with. Figure 3 (three left bars) shows a comparison of SVM with nearest neighbor and the prior baseline. We use 1,000 bootstrap samples and find the 95% confidence interval to be ±∼0.1%.

Looking at interpretable rules from decision trees, we noticed that a preference for the thickest or thin lines is used to predict a preference for thicker lines. Same for sparse vs. dense patterns. These preferences along a linear ordering are obviously (and hence uninterestingly) related. To verify that our models are not relying primarily on these uninteresting correlations, we reduced our set of Strokes preferences down to 5. We removed the sparsest, thickest lines and thin lines alternatives from Figure 2 because those were the least preferred alternatives along the thickness and density parameters. We retrained our models. Their performance is shown in Figure 3 (right two bars). We see that while model accuracies go down a little, they continue to be significantly better than the prior baseline. This suggests that we can indeed predict meaningful dependences between a user’s preferences when interactively creating generative art (Strokes).

Conclusion and Future Work

Future work includes expanding the study to more configurations and generative art forms, and translating the ML models to a smart interactive tool. This leads to more complex ML problems of modeling the sequence of interactions, and determining if the models should be used to eliminate part of the parameter space or promote a part of the parameter space. This involves focussing on either precision or recall of the models. Grounding this study in models of aesthetic experience (Leder and Nadal 2014) is future work.

To summarize, while we have not yet found evidence for it, given the narrow scope of our study, there may still be potential in the use of interactive generative art creation as an engaging and creative “personality test”. In fact, it may be possible to design generative art that explicitly optimizes for correlation with personality traits or preferences in other walks of life. We do find evidence that preferences of a user creating art using an interactive generative art tool are predictable from choices they make. This opens up opportunities for smart casual creators that make it easier for a lay person to create a piece they are personally excited about!

References

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2013. Designer Modeling For Personalized Game Content Creation Tools. In AIIDE.
Do Digital Agents Do Dada?

Gunter Loesel
Zürich University of the Arts
Zürich, Switzerland
gunter.loesel@zhdk.ch

Piotr Mirowski and Kory W. Mathewson
Improbotics
London, UK and Montréal, QC, Canada
contact@improbotics.org

Abstract
Do digital agents do Dadaism? To answer this question, we review a series of theatrical experiments involving human improvisers and AI-controlled Cyborgs in front of audiences. We describe these experiments and discuss the use of conversational digital agents (DA) on the stage. We identify two basic strategies of staging machines: the “immersive approach”, and the “Dadaistic approach”. We draw on Dadaistic conceptions of embracing modernity specifically in the Dadaists’ obsession with androids and cyborgs. Through analysis of several stage experiments we contend that digital agents, while attempting to build believable characters, do Dada, only if we do Dada too.

Motivation: Improvisation and Digital Agents
Improvised theatre, *improv*, is an art form modelled on natural human interaction which demands constant adaptation to evolving contexts. Previous research has paralleled *improv* theatre and jazz music and formulated both as “real-time dynamical problem solving” (Bruce and others 2000; Johnson-Laird 2002; Magerko and others 2009).

The first problem is collectively creating stories by impersonating believable characters and incorporating narrative elements suggested by the audience. The second problem is how to be truthful and in the moment of the scene, while accepting the offers made by the other improvisers, the audience, or their own cultural background (Johnstone 1979; Merritt and Hines 2019). The third problem is finding the limits of an audience’s expectations.

Logic in Improv and the Circle of Expectation
Improvisers follow the rules of logic while establishing a specific universe in which an improvised scene takes place. In this way, the performers and the audience can follow the story, and can predict how the scene continues (Merritt and Hines 2019). Practitioners introduced the *Circle of Expectation* as a concept to qualify the difference between adding obvious narrative elements that make the story more specific versus those that violate the expectations of the audience (Johnstone 2014). The circle contains the assumptions and associations that define the dramatic world (Mathewson and others 2020). Improvisors make offers by modelling obvious next-steps from the mind of the audience. Improvisers first establish the who, what, where, when, and the relationship between the characters, and then “do the most obvious thing” to move the story forward (Johnstone 1979).

Digital Agents in Improvisational Theatre
For digital agents (DA) based on randomness, adhering to logic in improv, while staying within the Circle of Expectation, seems an impossible task. Nevertheless, the field has adopted technological trends continues to innovate toward this goal. Recent work builds upon computational improvisation in music and dance performance (Fiebrink 2011; Hoffman and Weinberg 2011; Long and others 2020), and collaborative storytelling (Perlin and Goldberg 1996; Hayes-Roth and Van Gent 1996; Riedl and Stern 2006; Magerko and others 2011).

There have also been attempts at generating high-level narrative consistency for improvised theatre. DAs have acted as directing narrators grounding the performances of human improvisers with generated plot points (Eger and Mathewson 2018). Other DAs have quantified and shaped
the narrative arc of generated text in interactive human-machine dialogue, in order to reveal or conceal information according to the Circle of Expectation (Mathewson and others 2020). DAs have been used elsewhere in improv theatre (Bruce and others 2000; Baumer and Magerko 2010; O’Neill and others 2011; Knight and others 2011; Jacob 2019) and Simone’s Bot Party: Improv Comedy with Robots.

Progress in machine learning for natural language processing, specifically neural networks for text generation (Vinyals and Le 2015; Radford and others 2019), encouraged practitioners to build DAs for improv by focusing on the conversational and storytelling aspects. Mathewson and Mirowski innovated in this space with their collaborative human-robot improv group HumanMachine.1 They developed A.L.Ex., an advanced conversational chatbot trained on film dialogue from OpenSubtitles (Tiedemann 2009) which interfaced with speech recognition, synthetic text-to-speech, and controlled a humanoid robot stage partner (Mathewson and Mirowski 2017a; 2017b).

Advancing Digital Agents for the Stage

This study focuses on AI improv, where the robot is replaced by a human, who performs lines provided by an AI chatbot. Examples of such shows include Improbotics2 by Mirowski and Mathewson, Yes, Android by Etan Muskat and Almost Human3 by Gunter Loesel. In Improbotics, a DA sends lines of dialogue to a human performer Cyborg. While the Cyborg is free to move and to express non-verbal acting and emotional subtext, they can only say AI-generated lines. Those lines of dialogue are generated in response to context typed by a human Operator who also serves as curator in the case when multiple choices are offered.

The AI plays the role of an interactive playwright, giving lines to a specific performer, while challenging the other improvisers to justify the potentially nonsensical lines of dialogue. Experiments have investigated what kinds of theatrical frames emerge through the interaction of humans and machines on the stage, how one can describe and explain these theatrical phenomena, and what value these partly machine-generated dialogues have in an artistic and aesthetic sense (Martin and others 2016; Mathewson and Mirowski 2017b).

Previous work uses artistic lenses to contextualize novel AI technology (Horswill 2012; 2016). We analyze DA improv through the lens of Dadaism, which, we argue, is a suitable artistic frame to understand human-machine co-creativity improvisation.

Avant-garde Movements and Dadaism

The avant-garde movements at the beginning of the 20th century were the first artistic response to modernity and the industrial age. As the art historian Matthew Biro points out, the Dada movement was quite obsessed with the figure of the cyborg (Biro 2009)—though the word “cyborg” did not exist at that time. Dada artists, especially in the Berlin Dada group, explored and involved machines in various ways to express their ambiguous relationship towards modern life. They were furious activists against the political and military “machinery” that led to the first world war. For example, Hausmann’s drawing “Der eiserne Hindenburg” (1920) displays the German general Hindenburg as half-human, half-machine indicating the de-humanizing impact of technological war and chauvinistic patriotism.

Not only did Dada artists point towards the political aspects of automatization, they also reflected on the human self turning more machine-like. In two famous self-portraits, George Grosz’s “Daum marries her pedantic automaton George” (1920) and Hausmann’s “Selfportrait of the Dadasoph” (1920), the Dadaists displayed themselves as half-machine and half-human. These works explored a new relationship between human and machine. The humans had no outside perspective to look at the machine as a human. They are already changed and turned into cyborgs and are unable to distance themselves from the machine.

This reflection is explored in (Hausmann 1921): “Why can’t we paint pictures today like those of Botticelli, Micheangelo, Leonardo, or Titian? Because human beings have completely changed in terms of their consciousness. This is the case not simply because we have the telephone, the airplane, the electric piano, and the escalator, but rather because these experiences have transformed our entire psychophysical condition.” The naïve perspective of the human as a counterpart of technology was questioned even at this early stage of reflecting on modernity. Consequently, the Dadaists proposed an art not made by humans. Grosz stated this clearly at the First International Dada Fair in 1920, when he declared: “Art is dead. Long live the new machine art of Tatlin” (Broeckmann 2016).

Dadaism has a couple of identifiable traits. One of particular interest is that the artists deconstructed meaning, arriving at the smallest units of expression—letters or phonemes—resulting in poems that consist of single letters only (Hausmann, “fmsbvy”, 1918). This results in the Dadaistic practice of reducing the world to its basic units. Then putting those pieces together again in a nonsensical way, thereby uncovering the meaninglessness under the smooth veneer of sense-making. Optophonetic poetry (i.e. visible speech sounds) is an early example of the ambiguity of cyborg art. While subverting the meaning of language, it was performed in an expressive way, as can be experienced through recordings with Hausmann reading his own poetry.4

Dadaists did not embrace the machine age. They squeezed it so hard that modernity stopped being something you could look at from the outside. It became a feature internalized by modern human beings.

Artistic Strategies for Cyborg Improv

In appearance, trying to build a DA that acts like a human seems superfluous artistically. Even if it were possible, which at the moment it is not, such an endeavour would only mimmick human-to-human interaction. Quoting (Turing 1951): “But, I certainly hope and believe that no great efforts

---

1. https://humannmachine.live
2. https://improbotics.org/
3. https://www.stupidlovers.de/de/show-termin/almost-human
4. https://youtu.be/2lVqiCURmFQ
will be put into making machines with the most distinctively human, but non-intellectual characteristics such as the shape of the human body; it appears to me to be quite futile to make such attempts and their results would have something like the unpleasant quality of artificial flowers."

DAs can, however, be employed to create new stage interactions. The benefits owing more to the DA's deficits in social and psychological abilities than to their high-fidelity imitation. We now analyze digital agents in the context of such interactions.

Subverting the Turing Test for Deception
One component of AI improv shows consists of performing a few scenes while hiding the identity of the actor(s) who get their lines from an AI chatbot. All improvisers wear head-phones, whether they hear the AI or not. The performers are asked to improvise a poem based on a word suggested by the audience, one line at a time, and after a few verses, the audience is asked to identify the theatre in which creators aim to immerse the performance. One line at a time, and after a few verses, the audience is asked to identify the identity of theCyborg. This game is based on the Turing Test or Imitation Game (Turing 1950), where the machine needs to convince an audience of human judges that it is human. This dynamic is explored in other games (Treonor and others 2015) including Spy Party.5

In order to understand and to quantify the performance of humans and AI models alike, Mathewson and Mirowski (2018) collected feedback from a large number of human performers who interacted with AI co-stars. The human performers highlighted the limitations of the DAs; agents generated sentences inconsistent with respect to logic, social conventions, and emotions. Interestingly, these limitations were welcomed by the performers as they felt the AI acted like an "X factor" and forced humans to become better improvisers.

Anecdotal evidence suggests that, while performing the Turing Test, the actors naturally introduced a slight modification to the principles of the game.6 Not only was the Cyborg trying to pass as human, the humans tried to pass as robots. The non-Cyborg performers tried to behave like the Cyborg improvisers, aping the nonsensical nature of AI-generated text. This artistic choice was made by the human performers to deceive the audience into thinking that they actually were the Cyborg. The audience was fooled by the behaviour of the human actors, to great comedic success.

Based on feedback and observation, the actors gave the impression of not being "in their heads" while coming up with AI-sounding nonsense, rather, this nonsense was produced naturally. By being robotic, the actors behaved humanly upholding the audience’s expectations.

Cyborgian Dadaism
Contrasting the setup above, Almost Human (Zurich and Germany, 2018) used an alternative approach. In Almost Human, the machine did not pretend to be a human. The technology was provocatively not hidden. The show questioned the "machineliness" of the human.

This change of perspective opposes mainstream “immersive” theatre in which creators aim to immerse the perform-

<table>
<thead>
<tr>
<th>IMMERSIVE</th>
<th>DADAISTIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hiding technology</td>
<td>Displaying technology</td>
</tr>
<tr>
<td>Celebrating humanity</td>
<td>Celebrating the Cyborg</td>
</tr>
<tr>
<td>Mimic human behaviour</td>
<td>Expose machine behaviour</td>
</tr>
<tr>
<td>Within Circle of Expectation</td>
<td>Break Circle of Expectation</td>
</tr>
<tr>
<td>Try to pass Turing Test</td>
<td>Inspire the Cyborg in us</td>
</tr>
</tbody>
</table>

Table 1: Approaches for staging Digital Agents in improv.

ers and spectator in the fiction, making them forget the technology being used. Thus, there are two antagonistic strategies of dealing with modernity which we call the “immersive approach” and the “Dadaistic approach”.

Almost Human staged two chatbots in leading roles in a theatre piece—to our knowledge this is the first time this was done in a feature-length theatre performance. One chatbot was embodied by a human actor who got their lines from an in-ear-monitoring system (similar to Improbotics), and the lines came from JANN, the Just Approximate Nearest Neighbor chatbot.7

For the show, JANN was trained on a large corpus of pop-song lines in order to make it respond concisely and emotionally. The second chatbot was a modern version of ELIZA, the chatbot therapist (Weizenbaum 1966). ELIZA was embodied through a real-time avatar, projected on the screen (see Fig. 1). The audience was informed which lines came from a machine, so there was no confusion about who was human and who was machine. The show’s creative team made it explicit that the chatbots were co-creators of the theatre performance.

Often times the digital agents added to the story, other times they acted unpredictably and added absurd and illogical material to the scene. By letting the audience know the responses were created by an algorithm, they were “invited in” on the conceit of the show. They knew that the humans did not have complete control; the scene could move in unforeseen directions due to the contributions of JANN or ELIZA. At the same time the actors reacted emotional to all lines from the machine, providing a sense of urgency and drama. The audience could watch a play that was generated partly by a machine and partly by humans, making it “cyborg art”. The intention was to leave the audience with some irritation about who was the author of this performance, if it was a machine or a human—or if there was an author at all.

Discussion and Conclusion
A transfer of Dadaistic principles to today’s improvised performances is possible and consistent with the avant-gardes’ strategies for dealing with modernity. The use of AI in art has been heavily inspired by the Turing Test—training machines in their ability to imitate humans. Drawing on the avant-garde gives us a second frame for appreciating artistic human-machine collaboration. These theatrical experiments use two strategies. First, an immersive strategy that hides technology from the spectators and creates insecurity about the humanness of the actors. Second, a Dadaistic strategy

5 http://spyparty.com
6 Recording of an AI improv show: https://youtu.be/bMiSigawTu3s
7 https://github.com/korymath/jann
that displays technology and invites the audience to reflect about their machineliness and humanness.

The Dadaist deconstructed meaning, they concerned themselves with nonsense and irrationality. The anti-logic and anti-reason mentality of Dada broke open the Circle of Expectation. Dadaism defined itself by breaking those expectations and by shocking audience with seemingly dis-connected associations.

In summary, we provide several conclusions supporting the continued creation of AI improv using the frame of Dadaism. The Dadaistic strategy has a logic that integrates probabilistic components. The immersive approach is over-represented in popular culture. Much of the computational text generation underlying AI improv involuntarily frees the language from semantics and syntax. This quality aligns well with the fundamental underpinnings of Dadasim.

We suggest to use the Dadaistic strategy as a measure of the “realism-absurdity degree” of an improvised scene. The optimal balance of absurdism and realism for artistic purposes remains an open question. Most previous work on AI improv focused on the justification of what the AI would say, viewing the machine-generated nonsense as a challenge to good improv. We encourage investigation of how the absurdity of AI-generated dialogue uplifts improv. Finally, we conclude that DA do Dada, only if we do Dada too.

References
Horswill, I. D. 2012. Punch and judy ai playset: A generative farce manifesto, or, the tragical comedy or comical tragedy of predicate calculus. In AIIDE.
Horswill, I. D. 2016. Dear leader’s happy story time: A party game based on automated story generation. In AIIDE.
Neuro-Symbolic Generative Art: A Preliminary Study

Gunjan Aggarwal\textsuperscript{1} and Devi Parikh\textsuperscript{2}

\textsuperscript{1}guaggarw@adobe.com \textsuperscript{2}parikh@gatech.edu

\textsuperscript{1}Adobe \textsuperscript{2}Georgia Tech and Facebook AI Research

Abstract

There are two classes of generative art approaches: neural, where a deep model is trained to generate samples from a data distribution, and “symbolic” or algorithmic, where an artist designs the primary parameters and an autonomous system generates samples within these constraints. In this work, we propose a new hybrid genre: neuro-symbolic generative art. As a preliminary study, we train a generative deep neural network on samples from the symbolic approach. We demonstrate through human studies that subjects find the final artifacts and the creation process using our neuro-symbolic approach to be more creative than the symbolic approach 61\% and 82\% of the time respectively.

Introduction

Generative art refers to art generated using code, and typically includes an element of chance. Interactive versions of these systems can be viewed as Casual Creators (Compton and Mateas 2015). There are two dominant approaches for generative visual art. The first uses deep neural networks to generate images from a distribution that mimics training data. Indeed, generative artists train models on specific photographs they take or collect (e.g., Helena Sarin, Robbie Barrat) or perturb the weights of models to create artistic “glitches” in the generated art (e.g., Mario Klingemann). Another source of control is the random noise input to the model. Interpolations of two noise vectors smoothly control the generation in a local neighborhood.

In the second approach an artist defines an algorithm to generate art. An autonomous system generates random samples using this algorithm. Early algorithmic artists include Georg Nees and Vera Molnar. Algorithmic art is often abstract, with geometric structures, or repeating or recursive patterns. These “symbolic” approaches typically have explicit parameters to control the generated art.

To the best of our knowledge, these two approaches to generative visual art – neural and symbolic – have been largely distinct. This work is a preliminary study in exploring their intersection: neuro-symbolic generative art. Specifically, we train a Generative Adversarial Network (GAN) on samples generated using a symbolic approach. We hypothesize that the organic, unpredictable aesthetic associated with neural approaches complements the crisp, designed aesthetic of symbolic approaches. Moreover, compatible with data-hungry deep models, symbolic approaches support generation of large amounts of training samples. Example generated art samples from our approach are shown in Figure 1. Our human studies show that subjects find the artifacts and the interactive creation process using the neuro-symbolic approach to be more creative 61\% and 82\% of the time respectively compared to the symbolic approach.

Related Work

Neural generative models. These include Generative Adversarial Networks (GANs) (Goodfellow et al. 2014), Autoregressive Models (Salimans et al. 2017), Latent Variable Models (Kingma and Welling 2014), etc. Recent progress in GANs enables realistic natural (Brock, Donahue, and Simonyan 2019) and high resolution human face (Karras, Laine, and Aila 2019) image generation. We limit our study to GANs. GANs to generate video game levels (Giacomello, Lanzi, and Loiacono 2018) are particularly relevant as neu-

Figure 1: Example neuro-symbolic generated art samples.
A user can control the samples via the input noise vector, and interpolations of two noise vectors. For interpolated generation, we sample two noise vectors and then create arbitrary linear interpolations between the two vectors. Neuro-Symbolic Interpolations (NSI) are then generated by feeding the model each of the interpolated latent vectors. Specifically, NSI $x$ is generated from the generative model $G$ as

$$x = G(z); z = z_1 + \alpha \ast (z_2 - z_1)$$

$z_1 \sim N(0, 1); z_2 \sim N(0, 1); \alpha \in (0, 1)$ set by the user.

Figure 3 shows example interpolated samples.

**Human Evaluation**

We perform evaluation of both the artifact and the user-driven interactive creation process via human studies on Amazon Mechanical Turk (AMT). Subjects were from the US, with an AMT approval rating of 95% or higher, and having completed at least 5000 tasks on AMT in the past. They were paid above federal minimum wage.

**Artifact Evaluation**

We compare three artifacts – Symbolic, Neuro-Symbolic Generation (NSG), and Neuro-Symbolic Interpolated generation (NSI). These replicate the different artifacts a user might create when using the symbolic or neuro-symbolic interactive generative art tools. Human subjects were shown a pair of art pieces, one each from random two of the three types. They were asked which piece 1) seems more different from art you’ve seen in your life? 2) looks better? 3) is more creative? 4) is more artistic? 5) seems more likely to be hand-made? Subjects were also asked to optionally state why they felt one art was more likely to be hand-made than the other. The study consisted of 60 pairs, equally distributed across the three pairs of approaches. The study was completed by 20 unique subjects, resulting in a total of 1200 pairwise assessments.

Note that if we paired arbitrary pieces from two approaches, more than just the style of the art would likely differ (e.g., the color palette). To control this, the pairs were formed by finding nearest neighbors across approaches using color histograms. This helps minimize unrelated variations, and helps focus the study on the different styles of samples. Example image pairs shown to subjects can be found in Figure 4. Specifically, we first generate 10k NSG samples. We then pick a pair of samples and compute an NSI sample associated with the pair using $\alpha = 0.5$. We generate 10k such NSI samples. Recall that we already have 10k Symbolic samples in our dataset. Now to form a NSG vs. Symbolic pair, we pick either a random NSG or Symbolic sample from our dataset, and find the nearest neighbor from the pool of 10k images of the other category. Same for NSI vs. Symbolic, and NSI vs. NSG. The two images in a pair are randomly shuffled before showing it to subjects.

As quality control, we additionally asked subjects the number of colors in one of the artifacts in a pair. The number of colors in the symbolically generated art is known, giving us a way to identify subjects not doing the task well. Beyond 1 through 5, we gave subjects an added option of “Shaded colors, so not meaningful to count.”
The proportion of times users preferred one art form over another is shown in Figure 5. A one-sample proportion hypothesis test suggests that for our sample size, a “win ratio” over 0.54 (or below 0.46) is statistically significant at 95% confidence. These are shown as a horizontal lines in the figure. **Novelty, unusualness:** We find that the human evaluators rate NSG and NSI as being more “different” from art they’ve seen before than the Symbolic art about 66% of the time. **Better quality, value:** Subjects like NSG, NSI and Symbolic almost equally. **Creativity:** The third and fourth dimensions (“creative” and “artistic”) focus directly on the creative aspect of the artifacts. We see that human subjects find NSG and NSI to be more creative than Symbolic art about 61% and more artistic about 63% of the times. Note that NS(G/I) and Symbolic rated similarly for quality, but NS(G/I) were rated higher for novelty. We hypothesize that this results in NS(G/I) being rated higher in creativity overall (novelty + value, (Boden 2004)). **Naturalness, hand-made:** Subjects find NS(G/I) art to be more natural or more likely to be hand-made. Based on the comments shared, while certain subjects preferred Symbolic art as being more hand-made because of “perfect coloring”, about 59% of them chose the NS(G/I) art to be more likely to be hand-made because it “Looks like human error with paint dripping on to another color”, “The other piece of art has solid colors, where as the one I picked has various shades in spots.”, ”mixture of color together”, “smudge”. Finally, we see that NSI is preferred over NSG for novelty and creativity. **Creation Process Evaluation** Next, we evaluate live interactive generative art tools based on symbolic and our proposed neuro-symbolic approaches. Recall that the symbolic approach has 2 controllable parameters: color palette and the number of colors (maximum 5), as well as an option to generate a new variant of the art with the same parameters by changing the random seed. The neuro-symbolic tool has one controllable parameter $\alpha$ which generates an NSI between 2 NSG pieces, and an option to sample a new pair of NSG art by sampling new noise vectors. Human subjects were given links to both tools (Symbolic: http://genart.cloudcv.org/symbolic, Neuro-symbolic: http://genart.cloudcv.org) and for each, were asked to: “Find an art piece that you like a lot and share it with us!” and describe “what characteristics of your favorite art made it stand out from others?” Additionally, subjects were asked which tool 1) generates better looking art? 2) generates more surprising / unusual / unpredictable art? 3) generates more creative art? 4) is more satisfying to work with? 50 unique subjects participated. Half were given the symbolic tool first, and the other half the neuro-symbolic tool. Both tools had an option to add up to 5 pieces to their “favorite” gallery so users can keep track of pieces they like as they encounter them. Users could delete pieces from the gallery to replace them with others. They were provided an easy way to copy the URL of their favorite piece and submit it.

The proportion of times subjects preferred the neuro-
symbolic (NS) tool over symbolic (S) is shown in Figure 6. Trends are similar to artifact evaluation. **Better quality, value:** They like art generated by both tools equally. **Novelty, unusualness:** Subjects rate NS to be more surprising and unusual than S 68% of the time. **Creativity:** Subjects find the NS tool to generate more creative art than S 82% of the time. **Satisfying:** Interestingly, while less creative, subjects find S to be more satisfying to work with (albeit, not with statistical significance). An indicative comment from a subject: “I liked the task. I found that in [NS] the colors felt like they mixed together more. I found that in [S] was more clean looking and that made it more satisfying to work with in my opinion.” Using S is perhaps more analogous to “zen” (relaxing) activities, while NS may be closer to cognitively taxing creative activities. Exploring this is future work. Other comments about the two tools: “I had a bit more creative control with [S], while [NS] did generate more interesting combinations, it was just harder to get there predictably.” “[NS] provided more creativity, versus taking out colors like in [S].” “I liked in [S] the ability to choose the number of colors. I felt that in [NS] it was a little harder using my mouse to get the form and shape of the circles I wanted.” “This was a very interesting experiment, especially [NS]. I kind of felt like I didn’t know what to expect when I was trying to make my hybrid art.”

Example generated art samples beyond those shown in Figures, screen captures of our interactive generative art tools, and “favorite” pieces created by subjects along with a description of why they like the pieces can be found here: [https://sites.google.com/view/neuro-symbolic-art-gen](https://sites.google.com/view/neuro-symbolic-art-gen).

**Conclusion**

We present a preliminary study on neuro-symbolic generative art. It combines what have typically been two distinct approaches to generative visual art: neural and algorithmic/symbolic. We trained GANs on data generated via a symbolic approach. We evaluate the generated art and build live interactive generative art tools using both approaches. Human studies show that subjects find the neuro-symbolic generated art and creation process to be more creative than symbolic counterparts 61% and 82% of the time respectively. Overall, we see promising indications that neuro-symbolic generative art may be a viable new genre.

**Future Work.** We will explore other symbolic art styles, and train a model over multiple styles to potentially discover entirely novel styles. We further plan to interpolate between two symbolic images instead of two neuro-symbolic images. For this, we will explore techniques that map real images to latent representations. A user can then first design the two ends points (symbolically), and then generate an intermediate piece (neurally). Neither symbolic nor neuro-symbolic approaches alone allow for this level of control. Finally, training GANs directly on symbolic representations is an interesting and open research question.

**Acknowledgment.** Abhishek Sinha for helpful discussions.

**References**


Creative Constellation Generation: A System Description

Andres Sewell, Andrew Christiansen, Paul Bodily
Computer Science Department
Idaho State University
Pocatello, ID 83209 USA
{seweandre, chrian10, bodipaul}@isu.edu

Abstract
The discovery of new patterns in the sky and the attribution of names to those patterns have thus far been an exclusively human act. We believe that this makes constellation generation a great candidate for a computationally creative system. Artistically speaking, the night sky can be viewed as an infinite canvas and our system leverages that fact to produce novel constellation generations without prior knowledge of existing constellations. Given an image of stars, our system discovers a pattern, identifies an object in the pattern, and assigns it an intentional and creative name based on the object that the pattern resembles. We argue that our system exhibits creativity by emulating the same process by which a human would discover a new constellation. We describe how the system works, analyze the reception that its generations have received, and discuss the implications and future work that could be done to improve the system.

Source: github.com/DrewTChrist/CANIS/

Introduction
The Constellation and Name Invention System (CANIS) is a computationally creative system for generating constellations among images of stars. Finding and naming new constellations is one of the oldest expressions of human creativity. For thousands of years, different cultures and civilizations have each developed their own set of constellations. Currently, the International Astronomical Union recognizes 88 official constellations. In any case, compared to other domains of creativity, the field of constellation creation is unique for its longevity, its universality, and yet its relatively small set of available artifacts.

We define a constellation as a fixed group of stars to which a definite name has been given. It follows that a constellation must consist of two discrete pieces: an interesting or recognizable pattern of stars that collectively bear a resemblance to an animal or object, and an associated name that is somehow relevant to the resembling animal or object. The system generates novel constellations by following a linear set of steps: image processing, pattern finding, pattern matching, and intentional naming. None of these steps are new or revolutionary when considered individually, but we argue that our system’s creativity lies within the combination of these concepts to mimic the process that a human would use to find a new constellation. That is, by first finding an appealing pattern of stars in the sky, drawing from their preexisting knowledge-base of human experience to fit the pattern to an object or animal, and then assigning their discovery a memorable name to set it apart from other constellations.

Related Works
There are numerous works related to the problems that CANIS tries to solve, but they typically find themselves outside the domain of computational creativity. The list of related works involves those with somewhat overlapping astrometric and linguistic goals which will be discussed further.

Astrometric Works
The word astrometric refers to a branch of astronomy that measures the positions and separating distances of celestial objects. Many astrometric applications require the ability to efficiently match patterns. This particular problem is not new and several solutions have been formulated since the need arose (Groth 1986).

Linguistic Works
The demand for a quick, automated, and creative naming solution has motivated researchers to explore whether or not the concept can be made a reality. Analytic and relational linguistics projects such as ConceptNet and WordNet help to enable computational name generation, although much work remains before valuable and automatic name generation becomes a reality (Liu and Singh 2004; Miller 1995; ¨Ozbal and Strapparava 2012).

Computer Vision Works
Since its initial introduction in 1914, the Hausdorff distance function has been commonly applied to various computer vision problems. CANIS relies heavily upon the Hausdorff distance function for pattern-matching because it provides a useful metric for measuring object similarity within images (Rucklidge 1997).

Approach
Neither of the authors had experience working on a computationally creative system before CANIS, which in turn allowed us to exhibit our creativity for how we approached problems. Figure 1 is a high-level diagram that demonstrates how the system works. The objective is to take an image of stars as input, generate a novel constellation within the image, and present it to the user.
Knowledge-base

A knowledge-base enables a creative system to access information that directly affects its generations. To achieve the best results, our system should have access to a variety of example objects and animals to refer to while fitting constellation patterns. The more examples the system has in its knowledge-base, the better the system will be able to find a believable match and in turn a more valuable artefact. Our representation of topic examples consists of a label and a set of image coordinates that represent the outline of the topic’s silhouette. To make it easy to modify and add to the knowledge-base at will, the examples are stored in a database that CANIS accesses during its initialization phase. This database was constructed from images of 2D object silhouettes found online. After the topic data is retrieved, it gets used to create a local cache of the knowledge-base on disk.

Image Processing

To begin searching for patterns among stars it is necessary to first extract a list of coordinates representing each star in an image. To do so, an image is first converted to grayscale and then to binary based on a configurable per-pixel brightness threshold. This preserves the position of each star in the image above the brightness threshold while eliminating noise and dim stars. Each pixel in the processed image is iterated through and the coordinates of any white pixels are appended to a list. To avoid having multiple sets of coordinates that correspond to the same star, a check is performed on each new set of coordinates to ensure that they are sufficiently far away from the other coordinates in the list.

Pattern Generation

Real constellation patterns vary wildly, but seem to follow a few universal rules: patterns are made up of between 5 to 15 stars on average, the edges connecting patterns do not intersect, and patterns generally contain cycles. During pattern generation, a random subset of a random number of stars is selected from the full set and processed by a minimum spanning tree algorithm to find non-intersecting edges. This method alone can occasionally generate interesting patterns, but by definition, a minimum spanning tree contains no cycles which makes it insufficient for our use case. To remedy this, a simple algorithm randomly adds one to two cycles to the generated pattern to improve its authenticity while ensuring that no intersections are introduced. The final collection of nodes and edges is used to build an undirected graph that is used during the visualization stage to plot the pattern over the original image.

Topic Comparison

The utilization of Hausdorff distance as a fitness metric enables the system to numerically compare two arbitrary sets of coordinates. Hausdorff distance can be understood as the greatest distance between any point in the first set to the closest point in the second set. Therefore, the Hausdorff distance between two sets of coordinates is small when the points in each are close to the points in the other. This property makes it an effective metric for comparing constellation patterns to topic examples. It is particularly useful in this case since we store both constellation patterns and topic examples as lists of image coordinates. To minimize the Hausdorff distance and find the overall best-fitting topic, each topic example in the knowledge-base is scaled to and centered over the generated pattern. Each topic example is then rotated 360 degrees
with a new Hausdorff distance being calculated at each 15-degree interval. The topic, scale, and rotation combination that produced the overall smallest Hausdorff distance is selected as the best fit.

For a given image, performing exactly one iteration of pattern generation and topic comparison tended to produce mixed results in our testing. Although the system can fit any constellation pattern, in practice it seemed unlikely for an arbitrary pattern to achieve a small Hausdorff distance to a topic example. Since a smaller Hausdorff distance implies a better fit which implies a more believable generation, we attempt to remedy this by performing several iterations of pattern generation and topic comparison. After each new generation, the best fitting configuration is saved and replaces the overall best-fitting configuration if a smaller Hausdorff score is produced. Since the system is allowed to produce more generations, better fits tend to be found, and therefore a more valuable artefact is produced as output. The key is to perform enough generations for a relatively well-fitting configuration to be found, but not so many that the same optimized solutions are produced as output every time.

Name Generation
After a best-fitting topic is found, its label gets used as input to our name generation algorithm. The algorithm utilizes ConceptNet, WordNet, and Spacy to generate a unique name (Miller 1995). CANIS utilizes one of three templates when it comes to name generation: either a unigram, bigram or trigram is initially selected. The unigram template consists of either a direct hypernym or the object name itself. The bigram is constructed out of either the combination of a verb and the object name or an adjective and the object name. The trigram template is constructed as a combination of an adjective, a verb, and the object name.

Two approaches are used to gather semantically related words for constructing names. ConceptNet appeared to be the best resource for gathering direct hypernyms, while word vectors and WordNet are a good combination for gathering verbs and adjectives. Large sets of words of a certain part of speech can be sampled from WordNet and searched through to find the highest relational value to the object name.

Results
CANIS successfully generates novel constellations, but quantifying the value of its generations is challenging. It is intuitive for a human to decide whether or not they see value within a creative artefact, but difficult to translate their reasoning into measurable quality metrics. For this reason, we felt that producing a survey would be the best way to judge the quality of our system’s generations.

Survey
Our survey consisted of 8 generated constellations and 6 sentiment statements per constellation. Figure 2 gives an example of the constellation generations that participants were shown. Participants were asked to mark each statement that they agreed with and were required to choose at least one per constellation. Their submissions were anonymous and no demographic information was collected. Our objectives were to gauge the human-perceived value of the system’s generations and to pinpoint its weaknesses. The sentiment statements we used were:

1. The constellation’s pattern is interesting.
2. The animal or object the system chose believably fits the constellation pattern.
3. The constellation’s name is relevant to the chosen animal or object.
4. The constellation’s name is interesting.
5. The constellation as a whole is interesting or creative.
6. I disagree with all of the above choices.

We received 10 submissions and averaged the results to obtain an overall summary of participant sentiment.

<table>
<thead>
<tr>
<th>Survey Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statement 1</td>
</tr>
<tr>
<td>Statement 2</td>
</tr>
<tr>
<td>Statement 3</td>
</tr>
<tr>
<td>Statement 4</td>
</tr>
<tr>
<td>Statement 5</td>
</tr>
<tr>
<td>Statement 6</td>
</tr>
</tbody>
</table>

Table 1: Averaged survey results from 10 participants.

Analysis
The survey results are given in Table 1. Although the results were not what we hoped for, we recognize that a sample size of 10 individuals is not substantial enough to draw any definitive conclusions. Nonetheless, we feel that there are still some valuable insights to be extracted from the results.
Participants did not tend to believe that the system had believably fit a topic to a generated pattern. We suspect that this is due to a perceived lack of intentionality behind fitted topics. Official constellations typically include some transparent art overlaid as a means of assisting the viewer’s imagination in seeing what the constellation represents, but CANIS does not currently provide an equivalent.

A majority of participants believed that the generated names were relevant to the topic while less than half thought they were unique or interesting. Atypical names seem to be considered more creative. We received additional feedback which suggested that unigrams are perceived as significantly less creative than bigrams and trigrams. We feel that this criticism is fair and that an additional creative step, such as translating the generated name to Latin, would help to remedy this.

Final Discussion

We discuss the implications of CANIS among the computational creativity community, the authors’ contributions to the project, and ideas for future work to improve the project.

Implications About our survey results, there are things that CANIS performs well and acknowledge areas where the system could use improvement. Although not every generation was spectacular, CANIS occasionally produced some exciting and surprising results. Regardless of whether CANIS is a truly creative system or mere generation, we cannot dismiss the fact that the system introduces new and interesting ideas into the domain. Like many other systems within computational creativity, work performed on CANIS serves as a preliminary example of what can eventually be accomplished within the field. It also serves as inspiration for additional ideas that may be valuable or worth exploring.

Future Work In its current state, the system’s generations lack sufficient context to convey why a particular topic was chosen as the best fit. This makes it easy to dismiss its generations as uninspired. Generating a transparent representation of the chosen topic image to overlay onto the constellation pattern would allow the system to convey its reasoning and intentionality behind its topic selection. Our vision for how this might be implemented can be seen in Figure 3.

We provided CANIS with less than 25 topic examples during development. Expanding the system’s knowledge-base with additional topics would lead to increased artefact variety and better fitting topics. Increasing the number of examples that the system has access to should result in generations more comparable to those of a human.

A final consideration is to include metadata about each topic in the topic database. Similarly, CANIS could be extended with methods that enable it to gather deeper semantic knowledge about a given topic. In practice, this might look like a query to Wikipedia or some other common information source. This information would then be incorporated into our name generation algorithm which would result in better, more unique name generations. It would also help to build a stronger argument for the system’s intentionality.

Acknowledgments

We thank Dr. John Edwards for his assistance in finding an effective pattern-matching metric for our use case.

References


The Pokéerator is a generator of Pokémon names and descriptions, based on user input. The names are generated by blending words based on syllables or characters according to a bigram language model. An accompanying description is generated by filling a template with ConceptNet answers. This sentence is then used as a prompt for text generation with the GPT-2 language model which was finetuned on Pokédex entries. The evaluation of the generated Pokémon names shows that the names are not realistic, but appreciated and creative.

Introduction

While many computational creativity systems produce “art for art’s sake”, ever more systems are starting to focus on creativity in applied domains. These applications range from headline generation (Alnajjar, Leppänen, and Toivonen 2019; Gatti et al. 2016) over cover art for music albums (Cruz 2019) to mnemonic devices (Bodily, Glines, and Biggs 2019). Gaming is also one of the domains where computational systems, either autonomous or in a co-creation setting, are becoming popular. In addition to prominent examples like ANGELINA (Cook, Colton, and Gow 2017), which can generate full games on their own, a number of creative systems are focused on helping human content creators produce assets, resources and flavour text, i.e. text that fits well with the style of the game and adds to the depth of the story, but has no practical effect on its mechanics.

In the Pokémon universe, the role-playing video games where human trainers battle each other’s little “monsters”, i.e. Pokémon, both the Pokémon names and the entries of the Pokédex (an encyclopedia storing knowledge and trivia about every Pokémon) are prime examples of flavour text. The Pokémon universe looks like a good application scenario for a creative generator: the monsters’ names have a very distinct look and appearance, and are not random but related to the characteristics of the Pokémon themselves (Kawahara, Noto, and Kumagai 2018). Naming a new Pokémon is thus a creative task that requires both intelligence and knowledge of the domain. A creative generator for names and descriptions would be beneficial for the authors of the game.

Current Pokémon games let the user customise the name, gender and look of their playable character. Previous research shows that this type of customisation can result in higher player engagement (Ng and Lindgren 2013). With the Pokéerator, customisation could go beyond the playable character and expand to the first Pokémon a player receives.

This work aims at producing a personalised Pokémon name and description, starting from user-provided concepts. For the names, the generator aims at capturing the intuition behind the names of many monsters, i.e. blending two words together (e.g. “Snorlax” is a blend of “snoring” and “relax”). The descriptions are to reflect these characteristics by describing them further. Next to implementing the Pokéerator, this work aims at evaluating the quality of the output of the system.

Related Work

The Pokéerator is concerned with creative naming; Namelette (Özbal and Strapparava 2013) is an interactive system that tackles a similar problem: it can generate brand, company or product names. It creates neologisms from user input based on characteristics as well as phonetic similarities. Related information about the words are derived from ConceptNet (Speer, Chin, and Havasi 2017) and WordNet (Fellbaum 2010). These are blended together to create a new name. An n-gram language model (LM), trained on the words in the CMU Pronouncing Dictionary (Weide 1998) computes the phonetic likelihood of the name. Namelette can also perform latinisation of the name, by adding a latin suffix to the name. In many ways, Namelette works similar to the Pokéerator as they both rely on word relations, blend words to create new names and evaluate based on n-gram LMs.

JAPE (Binsted and Ritchie 1994) is a program for punning in a question-answer format. It creates puns based on schemata, descriptions as well as templates and uses WordNet data to create the puns. Even though our system does not aim at creating puns, it uses a similar syllable-merging process for Pokémon name generation and relies on templates for text generation.

Like the Pokéerator, Churnalist (van Stegeren and Theune 2019) aims at automatically creating flavour text for computer games. The system generates fictional newspaper headlines by feeding user input and related words into a headline database and replacing the subjects. Similarities are the usage of related words and templates; the Pokéerator however uses GPT-2 to produce somewhat longer texts.

The Patent Claim Generator (Lee and Hsiang 2019) aims...
at contributing to the sparsely explored field of “augmented inventing”, i.e. having the computer produce innovations. It generates patent claims using OpenAI’s GPT-2 model (Radford et al. 2019). Similarly to our work, the large pre-trained LM GPT-2 has been adapted, in this case to the field of patent claims, to be able to generate a particular type of text.

Method

User Input. In order to get the initial words for name and description generation, the user is first asked 8 “personal” questions requiring one-word answers (e.g. name, hobby, favourite animal/plant/food). The questions are intended for the user to build a relation towards their own “inner Pokémon”.

Word Creation. The user’s answers are the input for creating the new Pokémon name. To restrict the search space and keep computational costs low, the system starts by selecting two words at random. In the next step, the inputs are blended. First, the words are tokenised into syllables using the Natural Language Toolkit (NLTK) syllable tokenizer (Loper and Bird 2002). Then, a list is created by merging the first syllables of one word with the last syllables of the other word. This is done for all possible combinations. The longer word can never completely be part of the blended output as it would be too recognisable. However, the shorter word may because it could be only one syllable and thus be skipped. If both words are of equal length, they are both taken into consideration (e.g. [starfish, yellow] → [starlow, starfishlow, yelstarfish, yelfish]). In case both words only consist of one syllable, the merge is done on character level. The first letters of the first word until the first vowel are merged with the last letters of the second word starting from the first vowel. Moreover, a suffix chosen randomly from common Pokémon suffixes is added (e.g. [green, cat] → [gr-at, c-een] → [gratgon, ceenlow]).

Name Ranking. After generation, the system ranks the names to use the best one as Pokémon name with the help of a syllable-based and a character-based LM. Input words with more than one syllable are first split into syllables, then grouped into bigrams and evaluated with the LMs. If the original words had only one syllable, they are split into characters and grouped into bigrams for evaluation. We trained four LMs for evaluation: Two were trained on Pokémon names stemming from a dataset which contains information on the 802 existing Pokémon (Banik 2017), and two on the 133k English words contained in the CMU Pronouncing dictionary (Weide 1998). We used two sets to ensure that the generated name looks like a Pokémon but also seems like an English word. For each dataset, one LM was created on the basis of syllables, and one on the basis of characters. For each word in the datasets, we created bigrams and subsequently calculated the probability of each bigram using Naive Bayes with Laplace Smoothing. The probability of a generated word is calculated by multiplying the individual bigram probabilities. For example, \( P(\text{starfishlow}) = P(\text{fish})|\text{star}) \times P(\text{low})|\text{fish}) \). The probabilities from the Pokémon and the CMU dictionary are weighted: \( P(\text{starfishlow}) = 0.4 \times \text{PokémonLM} + 0.6 \times \text{EnglishLM} \). The weights were chosen based on an internal evaluation of about 20 examples. This ensures that the word is pronounceable and not completely alien from English orthography, while still considering the peculiarity of Pokémon names (e.g. “Exeggeut”, “Kakuna”). The generated word with the highest probability is returned as the name of the Pokémon. In the above example, this is “starlow”.

Description: Prompt for text generation. The description of a Pokémon in the Pokédex is usually a short text of up to three sentences, describing one feature or characteristic of the Pokémon. In this work, the description is created using OpenAI’s GPT-2 model and an input sentence. The input sentence is generated based on word relations and templates. One of the words that compose the generated Pokémon name is taken as an input to ConceptNet (Speck, Chin, and Havasi 2017) in order to retrieve related words. ConceptNet offers a number of related words as an answer to one query as well as so-called surface texts, i.e. sample sentences including both the input word and the output word, specifying the relationship of the words (e.g. “Something you find at [sea] is [a starfish]”). The offered related words are filled into templates. As there are different relations, we prepared multiple templates for each relation. To ensure proper grammar, the retrieved word needs to fulfil a part-of-speech (POS) expected by the template sentence, e.g. a template for the word relation “AtLocation” is “It likes to be at <AtLocation>.”, which expects a noun. In order to ensure the correct POS of the output word, the surface text is POS tagged. From the available word relations that satisfy the described requirements, a fitting one is chosen randomly and the input sentence is built. In the example of the Pokémon “Starlow”, the input sentence for the next step is “It likes to be at sea.”.

Description: Text generation. To generate the Pokémon description, the pre-trained LM GPT-2 is used. GPT-2 (Radford et al. 2019) is an unsupervised LM which has proved useful for different Natural Language Processing tasks, including language generation. We finetuned the LM on a dataset of real Pokédex entries. The 802 descriptions (about 1,600 sentences) were scraped from the Pokédex website. The previously created input sentence is used as a prompt for the generation. The model returns a description of 100 characters which is stripped off after the first three complete sentences. The final description is composed of these sentences and excludes the prompt sentence as it is rather simple, not particularly creative, and would introduce a lot of repetitions due to the limited number of templates. Any mention of a Pokémon in the generated description is replaced by the generated Pokémon name. Finally, the generated Pokémon with its name and description is displayed to the user. In our example, the generated final description would be: “Starlow continually molts the shell and discharges toxic spores. This Pokémon feeds on toxic gases”.

---

1Given the relatively small number of Pokémon, using only a Pokémon-based LM would result in low probability scores, due to the limited amount of syllable transitions that could be covered.

2https://www.pokemon.com/us/pokedex/
and toxins. Starlow is capable of swimming in the sea.”. Due to the small size of the training corpus, GPT-2 can easily be overfitted. It may return a description which partially matches one in the Pokédex corpus. To avoid this and ensure novelty of the output, the ROUGE-5 precision score (Lin 2004) is calculated, i.e. the amount of overlap of 5-grams between the generated text and the training dataset. If the ROUGE-5 precision is larger than 0, meaning at least one 5-gram was detected, the description is discarded. A new description will be generated until this requirement is met. 3

Preliminary evaluation
The generated names were evaluated in a within-subject study with 33 participants, recruited through convenience sampling4. Only participants that had already played Pokémon, excluding those having played the latest generation (Generation VIII), were able to do to the evaluation. This was done to ensure at least a basic level of familiarity with Pokémon. The evaluation consisted of an online survey.

Participants were presented with 4 original Pokémon names (random selection from latest generation) and 4 generated names. For the generated Pokémon names, 4 results that looked convincing (e.g. not presenting the errors mentioned in the Discussion section) were chosen for the evaluation. Participants were asked to classify which of the names were generated and which names were original. This tested how realistic the generated Pokémon names sounded in comparison to original names. In addition, participants were asked to rate the names on two dimensions: likeability and creativity. The two variables were measured using a 5-point Likert scale.

In a follow-up study, 26 participants participants were asked to interact with the Pokélator to generate their own individual Pokémon. We collected their impressions of the system, and checked if it could help them “unveil their inner Pokémon”.

Results and discussion
Evaluation results. Regarding the evaluation of Pokémon names, users can identify most of the generated names as such (68% accuracy on average). This gives an indication that the generated names are not similar enough to real Pokémon names, or that it was too obvious that they were constructed from two words. This led to the names being easily distinguishable and indicates that improvements on this front are needed. It is worth noting, however, that original names were often mistaken as generated by the participants (on average, only 44% of non-generated names are correctly classified as “original”), suggesting an important effect of familiarity that should be further investigated.

However, in the dimensions of likeability and creativity (Table 1), no significant difference (using a paired t-test on the average per-participant ratings) could be found. A potential explanation is that generated names are liked as much as (unfamiliar) Pokémon names - again suggesting a strong effect of familiarity - , and that the Pokélator could be reasonably successful at producing creative names.

<table>
<thead>
<tr>
<th></th>
<th>Original</th>
<th>Generated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Likeability</td>
<td>3.37</td>
<td>3.14</td>
</tr>
<tr>
<td>Creativity</td>
<td>3.39</td>
<td>3.20</td>
</tr>
</tbody>
</table>

Table 1: Average ratings of Pokémon names

Finally, from the users that could interact with the system, we collected some feedback. About 20% of the participants stated to have found their inner Pokémon.

Error analysis. During the name creation process, words are blended together and the final name is selected based on the likelihood of the syllable/character arrangement making up a real word. We did not consider that the original words and their n-grams have a higher probability as they occur in the training data. This leads to a higher probability of word blends containing a full original word.

Another issue is the overfitting of the GPT-2 model which can lead to (parts of) generated descriptions being copied to the output. On the one hand, the relatively small size of the training set can easily lead to overfitting of the GPT-2 model. On the other hand, shorter training can lead to descriptions which are further from potential Pokédex entries. Currently, this problem is tackled by using ROUGE as an ‘overfit detector’ that will trigger the generation of a new description.

Limitations. Currently, the syllables are extracted from the input words using the syllable tokeniser from NLTK. However, the quality of the output of this tokeniser varies greatly, limiting smooth syllable concatenations. In addition, the method for naming in this work limits the possible names to only blended words, whereas real Pokémon names are not always blended words (e.g. “Ekans” which is “Snake” in reverse). For the description, the first limitation is the dependency on answers from ConceptNet. Since it is a crowd-sourced database some words have limited sets of relations to choose from while others have questionable relations, e.g. among the <AtLocation> words for “cat” are “my dogs mouth”, “a hat that comes back” and “the Milky Way galaxy”. A second limitation is the use of hardcoded templates as it only offers a certain number of simple sentence skeletons to choose from. This can have an effect on the quality of the generated description. In addition, there is a low connection between the description and the name and subsequently the user as only one word from the user-given input is used for generation description. This limits the level of self-identification of the user with their Pokémon. As for the evaluation, the main limitations - apart from the lack of data on descriptions - are its size and the generalisation of the results. We hand-picked a limited number of names, and these are not necessarily representative of the output but rather contain the top percentage of generations.
Future work. In addition to a more extensive evaluation, which should encompass descriptions in addition to a larger number of generated names, there is room for improvement in the system itself. Further work could be focused on improving the name generation process. Instead of literally using the user’s answers, the name could be generated with synonyms of the answers. Another possibility is to use all the answers from the user and generate all possible combinations. These would lead to greater variability in the generated names, and might lead to better results. Besides that, other combinations of syllables could be tried out, e.g. the syllables of one word are placed in the middle the other, as happens in real Pokémon names such as “Exeggutor”.

In addition, the evaluation of the different syllable combinations could be improved, e.g. by also using a phonetic LM which could lead to more realistic sounding words.

Looking at the description generation, some issues can be found with the sentence generated with ConceptNet data. Future work could focus on checking the relations retrieved from ConceptNet for grammar and content plausibility.

Finally, more features could be added to the Pokéerator, e.g. the Pokémon type and suitable attacks. This would result in a more holistic and complex Pokémon generation.

Conclusion
We investigated how to develop a creative system that can generate a new Pokémon with a description based on user input. The resulting Pokéerator blends user-provided answers together, producing a Pokémon name, and uses their properties to generate a short description. The evaluation shows that generated names are not realistic, but seems to achieve similar levels of likeability and creativity as original Pokémon names. From the individual evaluation, about 20% of participants found their inner Pokémon and could identify with it. With further improvements, we hope the system could prove itself a useful tool to assist Pokémon game developers, or to extend the possibility of user personalisation in the next Pokémon games.

Acknowledgements
Pokémon names are copyright of Nintendo/Game Freak; no copyright infringement is intended.

References
An Approach for Text-to-Emoji Translation

Philipp Wicke
Creative Language Systems
University College Dublin
Belfield 4, Dublin - Ireland
philipp.wicke@ucdconnect.ie

João M. Cunha
CISUC, Dept. of Informatics Engineering
University of Coimbra
3030-790 Coimbra - Portugal
jmacunha@dei.uc.pt

Abstract

The task of translating text to images holds some valid creative potential and has been the subject of study in Computational Creativity. In this paper, we present preliminary work focused on emoji translation. The work-in-progress system is based on techniques of information retrieval. We compare the performance of our system with three deep learning approaches using a text-to-emoji task. The preliminary results suggest some advantages of using a knowledge-base as opposed to a purely data-driven approach. This paper aims to situate the research, underline its relevance and attract valuable feedback for future developments.

Introduction

Computational approaches to automatic illustration of text have long been a subject of study. Several methods have been explored to produce visual representations in the field of Computational Creativity. Some consist of systems for collage generation, e.g. Cook and Colton (2011), others employ visual blending techniques, e.g. Xiao and Linkola (2015), and generative adversarial networks (GANs) have also been used to synthesise photorealistic images that represent a given theme, e.g. Ni et al. (2020).

When it comes to using images to convey concepts or ideas, it is impossible not to mention emoji. According to a report released by Adobe in 2019, emoji continue to thrive – users surveyed admit to include emoji in text messages 49% of the time. Despite being mostly used as a way to make conversations more fun or lighten the mood, 94% of the surveyed users identify the ability to communicate across language barriers as one of the greatest benefits of emoji. Two different functions of emoji can be identified (Dürscheid and Siever, 2017): complementary (to accompany text) and replacement (to replace words). While the former has been often computationally explored for the development of emoji prediction methods, the latter has not been given the same attention. An exception is the Emoji Replacement function introduced in iOS 10. Nonetheless, this exception mostly addresses the replacement of single words. 

Related Work

Several related works have inspired the methods applied in the proposed translation system. Closely related to our text-to-emoji system is the one presented by Wicke (2017). The author creates and evaluates a system that can translate action words into sequences of emoji through the use of vari-
rous linguistic strategies (metaphor, idioms, rebus etc). In the empirical evaluation the author concludes that action word translations using the rebus principle (e.g. 🌱 🌿 for “believe”), metaphors (e.g. 🎈 for “luck”) or literal translations (e.g. 🌡️ for the action “to explode”) are being best understood and appreciated by human readers.

If our system is tasked to translate text to emoji, we also consider how humans are performing in such a task. In a study by Wicke and Bolognesi (2020), the authors had 300 concept words (e.g. dog, democracy, luck, family) translated by crowd-workers using Amazon Mechanical Turk. With 11 translations per concept from individual workers, the authors analysed the coherence of translations correlated to the concreteness and abstractness of each concept. The results of their study indicate the more concrete the concept, the greater the coherence and the more abstract the concept, the more emoji are being used and the more face emoji occur.

Another approach to concept representation using emoji is Emojinating (Cunha et al., 2019) – a system that uses visual blending of emoji to represent user introduced concepts. However, this system is of little use for our study as it considers one-word and two-word concepts.

Regarding text to emoji translation, we describe three ML-driven systems that we will compare our system against: SemEval System, DeepMoji and DangoApp.

SemEval System is described by Cöltekin and Rama (2018) and was a contribution at 2018’s International Workshop on Semantic Evaluation Task 2: Multilingual Emoji Prediction. This model is based on a One-Vs-Rest Support Vector Machine architecture and has been trained on 500,000 Twitter messages (tweets). The training input was one tweet to predict one emoji on the basis of 20 classes of emoji (the most common emoji on Twitter at that time). The requirement for one tweet one emoji prediction was given by the SemEval task and is an obvious restriction in a text-to-emoji translation. Yet, this system has been included in the study as the code is freely available, easy to implement and allows us to reproduce the model.

DeepMoji is a system for the detection of sentiment, emotion and sarcasm in text using emoji, implemented by Felbo et al. (2017). DeepMoji is also freely available. This system is trained on 1.2 billion tweets containing one of the 64 most common emoji on Twitter. The neural network architecture is comprised of: Embedding Layer → BiLSTM → BiLSTM → Attention Layer → Softmax. Again, this project does not aim to translate text-to-emoji directly but to reflect the emotional content of a tweet through emoji.

DangoApp The fourth system is Software Product by WHIRLSCAPE released for Android/iPhone as an App. The self-titled “Emoji Assistant” is a real-time emoji prediction app. It claims to also capture slang expressions and memes as a result of its Deep Learning architecture with RNNs providing a semantic space for cosine similarity measures. All information is provided on their website: getdango.com. As the code and the model are not available, we use the app to test its text-to-emoji translation capabilities.

Our Approach: InfoRet

The goal of our system is the translation of a sequence of words (text) to a sequence of emoji that can convey the same or similar meaning. Contrary to all the presented ML approaches use data that has text-with-emojis instead of text-to-emoji, we decided to distance our approach from those. Instead, we adopted the insights from Wicke (2017) and used the database of action-to-emoji mappings in combination with ConceptNet (Liu and Singh, 2004) and EmojiNet (Wijeratne et al., 2017). Our method is explained in the next subsection and as those techniques are mostly information retrieval techniques, we call the system hereinafter InfoRet.

Dictionary Creation

In essence, the proposed system is built around a dictionary of translations from words to emoji. The core part of the research is the constitution of this dictionary. It includes the following entries:

2. All entries from EmojiNet Wijeratne et al. (2017) – a machine-readable emoji sense inventory that maps Unicode emoji representations to their English meanings extracted. It consists of 12,904 sense labels for over 2,389 emoji. Within the labels, we perform a term-frequency-inverse document frequency (tf-idf) analysis to weight the most important emoji for each label. We access ConceptNet in order to extend the labels provided. For example, the label “dog” can be extended with “canine” or “puppy” using ConceptNet.

The idea behind the first addition of action-to-emoji translations has been explained previously, the second addition of EmojiNet needs a brief explanation. With 12,904 sense labels, we are provided with a great addition to the dictionary. Yet, the emoji that are linked to a label by the Unicode are often very similar, e.g. “animal” is a label attached to every emoji that depicts an animal. If we would just use the first label, we would lose important information such as “animal, dog”. Performing a tf-idf on those labels allows us to extract the label most relevant to its overall frequency. Now, we can greatly expand the labels if the label “dog” can also refer to “puppy” or “canine” using ConceptNet.

Translation

The system takes a sentence as an input. The sentence will be filtered for common stopwords (I, me, am, him, his etc) (Stone, Dennis, and Kwantes, 2011). For each word in the sentence, it is checked whether the word is similar (similarity checked here using Python’s difflib SequenceMatcher) to an entry in the dictionary. If there is a match the corresponding emoji will be stored.
Study on Emoji Translation

In this section, we describe a preliminary study that tests our approach and three ML-driven systems in a text to emoji translation task.

Experimental Setup

We compare the four approaches in a text-to-emoji task. Even though our ultimate goal is to translate text to emoji in short stories, we decided to test the systems with other types of text as well. Therefore, we picked three different types of texts: a tweet (as this is what most of the ML approaches have been trained on), a short story (as this would be the ideal text length to be translated) and a poem (to compare creative aspects in a different domain). The tweet is one by Donald Trump and one by Barrack Obama, the short story is “Appointment in Samarra” by W. Somerset Maugham and the poem is “Ozymandias” by Percy Bysshe Shelley. We ran each of the four systems on each of the four texts (sentence by sentence).

Results and Discussion

In Fig. 2 we can see the results of each system for the respective text. We will interpret the results for each system separately before we conclude with a comparison.

SemEval: This method is primarily focused on the labelling of one tweet with one emoji. We can see the limitations for translating full text in the results i.e. there are only three individual emoji used for the translation: 26x️, 3x️️, 2x️️. Even for its target domain (tweets), the system annotates both tweets with the same “tears of joy”-emoji, despite the fact that both tweets carry a different sentiment. As this approach seems to fail in its own domain, there is hardly any use in other textual domains such as short story or poems. As to be expected, the results indicate that a sentiment classification with only one label per sentence is too far from the system we need for text-to-emoji translation.

InfoRet: The results of the InfoRet system are the most diverse and will need the most interpretation. As we can see in Fig. 2, the system generates sequences between zero (Short story line 10, poem lines 9 and 13) and six (Trump tweet line 2, poem line 6) emoji long. This is a variability we do not observe for the other systems. The cases of zero emoji can be considered a failure of the system, which occurred three times. This also means that the system “keeps quiet” when there is no definite solution, something the other systems do not account for. We can observe some instances in which the InfoRet system suggests a much better translation than the other systems. For example in line 10 of the poem, the InfoRet suggests the “name badge” and “crown”, whereas only the DangoApp captures the word “King”, but not the naming aspect of the text. We can also see recurring tweets over multiple lines, e.g. the button as defined in Trump’s tweet’s translation as 🎯. This button can be observed over multiple tweets, which is a feature that the other algorithms do not provide. This might suggest, that InfoRet is more useful for long text translations.

DeepMoji: This method also does not aim to be a text-to-emoji translation but rather to express the sentiment of a sentence. Therefore, it is useful for sentiment analysis of a sentence, but not necessarily for a direct translation. We can observe a strong overrepresentation of the music, keys and notes emoji in the poem, even though none of this is related to the text. Investigating the Trump tweet we can see that the first line is comprised of tweets signalling the same ridicule sentiment as the tweet, whereas the second line presents the negative sentiment of the text and the last sentence reflects self-praise with emoji. As good as this system captures the
sentiment as bad can you infer any meaning of the underlying text.

**DangoApp:** The results of the DangoApp can be seen as somewhat in between the InfoRet and the DeepMoji. It captures sentiment, yet it also allows including more concrete emoji. We can compare some examples: The Donald Trump tweet includes (line 1) the ridicule with the laughing emoji, but it also includes a flag (not the North Korean one though). Line 3 shows the positive face emojis and a flash emoji to relate to the “powerful” in the text. Yet, we can also see how this system fails in the poem line 9, where the musical emoji do not relate to the text.

Overall, our symbolic/information retrieval approach seems to show advantages over the machine learning approaches. The three deep learning approaches show strong deficits as soon as we leave the Twitter domain, due to the fact that they were trained on Twitter data. In fact, comparing these approaches in an emoji translation task of longer texts can be considered unfair as they have not been built for this purpose. Nonetheless, as there is no dataset of stories, poems or longer texts that are translated to emojis – and it is unlikely such a database will soon be created – the InfoRet is most likely the best approach to serve as a domain-unspecific, generic model for translating text into emoji. Our system does not seem to have much use for communicative purposes, as using emoji for word-replacement is not very appropriate for written communication and has even been shown to increase reading time (Gustafsson, 2017). For artistic purposes, we consider that InfoRet has potential, especially when it comes to automatic text illustration.

**Conclusion and Future Work**

In this paper, we presented an approach for automatic text to emoji translation, based on information retrieval techniques. We tested the system with three different types of texts: tweet, short story and poem. We compare the results with the ones from three machine learning-based systems. Notably, our evaluation is of preliminary, subjective nature. This paper describes work in progress and, as such, a more thorough validation needs to be conducted. Notably, a similar approach was made public after we conducted our evaluation in Day et al. (2020 forthcoming). For future work, it will be highly valuable to compare our approach with this system once it has been published.

**Acknowledgements** This work is partially funded by the FCT - Foundation for Science and Technology, I.P., within the scope of the project CICUT - UID/CEC/00326/2020 and by European Social Fund, through the Regional Operational Program Centro 2020, and under the grant SFRH/BD/120905/2016.

**References**


Felbo, B.; Mislove, A.; Seggaard, A.; Rahwan, I.; and Lehmann, S. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proc. of EMNLP.*


Radford, W.; Chisholm, A.; Hachey, B.; and Han, B. 2016. : telephone:: person:: sailboat:: whale:: ok-hand:: or” call me ishmael”-how do you translate emoji? arXiv:1611.02027.


Wicke, P. 2017. Ideograms as semantic primes: Emoji in computational linguistic creativity.


Hand-Crafting Neural Networks for Art-Making

Erik Ulberg¹, Daniel Cardoso Llach², Daragh Byrne²
Computational Design Laboratory, School of Architecture
Carnegie Mellon University
Pittsburgh, PA 15232 USA

Abstract
A growing number of visual artists use neural networks in their practice. While these networks show promise as an art form, the lack of interpretability limits control to high level decisions based on observations. As an alternative, this research investigates the hand-crafting of network weights coupled with explanatory visualizations as a form of creative control over the internal and lower level processes. Two experimental tools were developed: one for parametrically generating first layer kernels and the second for editing multiple layers. These tools attempt to transform the hand-crafting of features into “crafting” in a richer sense by bringing network weights and visual materials into a tight feedback loop. The first author extensively engaged with these tools and these case studies serve to examine the affordances of internal interaction for art-making. The findings suggest that direct manipulation can be used intentionally and can yield insights into network representations, but that hand-crafting networks of greater sophistication would likely require a hybrid approach integrating data-driven methods.

Introduction
Neural networks show promise as a form of representation in art. However, the available tools for manipulating networks limit creative control to decisions around datasets, algorithms, and hyperparameters (RunwayML). While these tools are valuable for their ease of use, they do not fully leverage artists’ fine-grained knowledge of the construction of images. Interaction happens on the exterior of networks based on high level observations of input and output. The internal and lower level processes of networks are overlooked as a potential site of engagement for artists.

AI researcher Christopher Olah asked the provocative question: “What if we treated individual neurons, even individual weights, as being worthy of serious investigation? (Olah et al. 2020b)” Explainable AI (XAI) research has demonstrated that networks often contain a rich world of visual concepts within their intermediate layers (Olah et al. 2020a; Sharif Razavian et al. 2014). This suggests that the right tools could enable artists to closely engage with network interiors and compose with low-level building blocks.

Two experimental tools for hand-crafting network weights were developed to explore Olah’s question: one for parametrically generating weights in the first layer of a network and the other for editing multiple layers. These tools do not use machine learning to train network models. Instead, they explore what can be accomplished through close study and the hand-crafting of individual network weights.

In the context of neural networks, the term hand-crafted refers to features designed by humans rather than learned from data. In creative fields, hand-crafted is often associated with workmanship, or the subconscious dexterity emerging from intimate experience with a material (Pye 1968). The tools presented here seek to reposition the crafting of networks to be a form of workmanship. Editing weights and visual materials in a tight feedback loop is intended to create a deeper level of engagement and to unveil new co-creative possibilities.

Background
Understanding and Interacting with AI
Deep neural networks encode visual concepts using tens of millions of weights in ways that are not well understood. To maximize their usability, many AI-powered artistic tools leave network weights as a black box and focus on high level, external controls. Thus, the interaction paradigm for creative AI applications tends to fall under the umbrella of what Zhu et al. (2018) define as Observable AI (OAI). In OAI, users build a mental model of a network’s function through observation of inputs and outputs.

OAI stands in contrast to XAI where the goal is to shed light on the inner functioning of networks. Research in XAI often includes interactive visualizations of the flow of weights or renderings of visual concepts within networks (Smilkov et al. 2017; Olah et al. 2020a). These efforts have yielded some insight, leading to the prevailing wisdom that network layers contain progressively higher orders of abstraction.

On close inspection, Olah et al. (2020a) found that the early layers of a network trained on natural images built up recognition from simple concepts such as edge detection, color contrast, and lines to higher level features including corners, patterns, intersections, and shapes. This evidence points to the existence of universal building blocks at intermediate levels in networks. Practitioners often leverage these building blocks through transfer learning to avoid...
training from scratch on new datasets (PyTorch). Their existence also provides motivation to find tools for manipulating the low-level, internal processes of networks.

Low-level interaction with network weights is generally viewed as impractical, but forcing a confrontation challenges users to deepen their understanding of networks. Smilkov et al. (2017) argue that rapid, direct manipulation readily provides intuitions on how networks function and sought to demonstrate this through their popular web interface “A Neural Network Playground.” Similarly, the hand-crafting tools in this paper are meant to provide intuitions through open-ended interaction and internal visualizations. While explanations of internal processes of networks is less favored in creative practice, this project uses it alongside observation. The tools presented here combine elements of both XAI and OAI by relying on observations of a generative drawing system in tandem with visualization tools for exploring and directly editing the inner weights of networks. The hope is that combining stimulating feedback with the hand-crafting of network weights will lead to fruitful artistic interactions.

Hand-Crafting Networks

Hand-crafting is used sparingly with neural networks because of the relative success of learned features. A number of studies have compared the performance of hand-crafted features to those derived through machine learning. These studies found mixed results showing that hand-crafted features can reduce training time, but struggle to match the accuracy of learned features (Antipov et al. 2015; Zhang et al. 2020). Regardless of these results, quantitative measures of success are less relevant in creative practice. The purpose of this project is not to improve the predictive accuracy of networks through new tools for hand-crafting, but to apply the technique in an art-making context where open-ended interaction and creative control are valued.

Directly crafted weights have generally been overlooked by the creative community. One partial exception that reinforces this observation comes from “Neural Glitch” by artist Mario Klingemann (2018). In his project, Klingemann manually altered the internal weights of a generative adversarial network to probe its inner representations. His results retain some of the appearance of the original output, but with haunting effects. Klingemann’s work reveals the delicate balance of weights within networks and shows how even small changes can have a dramatic impact on the learned structures. The name itself, “Neural Glitch,” contains the prospects facing an artist attempting to edit network weights by hand. The expectation is that direct manipulation will be a search for interesting glitches rather than a series of intentional choices. This project attempts to challenge that boundary.

Hand-Crafting Tools

The tools developed as part of this work are called the Kernel Tuner and the Network Builder (Fig. 1). They combine an editable canvas, interactive visualizations, and the hand-crafting of weights.

The Kernel Tuner is a parametric tool for crafting a single layer of weights to extract basic features from a line drawing. Generally, the process of using it begins by making a drawing on the canvas to serve as the basis for evaluating sets of parameters. Next, the user adjusts the sliders within the interface to rapidly test different parameters. The canvas can also be rotated to see how tolerant the kernels are to variance. While these updates are made, the Kernel Tuner provides real time visualizations of how the network responds. Through this iterative process, the tool facilitates human-led jumps through the search space of possible kernels.

The Network Builder uses the kernels from the Kernel Tuner as its first layer and assists with the more ambitious goal of hand-crafting multiple layers of convolution and pooling. Since explanation becomes more difficult with multiple layers, the Network Builder also offers opportunities for understanding the network’s function through observation. It achieves this by plugging in the network as a reward function for a generative line drawing system.

The line drawing system operates iteratively on the cur-
rent state of a canvas by drawing or erasing marks of a few pixels in length. The algorithm has two options: it can start a new line or continue a line it is already drawing. Either way, it generates batches of random segments, tests how each segment changes the activation score, and then chooses the highest score. Through this method, the algorithm greedily maximizes the activation of the network. The system terminates when it can no longer find segments that sufficiently improve the activation score (based on a tuned threshold). Additionally, it uses a line end detector to inject options that connect to existing line ends. This small modification makes the system much more likely to draw meaningful shapes.

Typically, the process of using the Network Builder starts with the manual entry of kernel weights (with the help of functions for rotation, reflection, and shifting). Then, the generative algorithm is run to produce several sample outputs to evaluate the success of the weights. If the samples do not match expectations, the generative algorithm is run again until it reproduces an aspect that is not intended (such as premature stopping). The user can pause the algorithm to draw and erase on the canvas while observing changes in network activations. This helps to explain why the final activation score is not responding as expected. Following inspection of the network activations, the steps are repeated. This crafting process involves both observation (of the generative system’s response to a given canvas state) and explanation (through the tracing of network weights).

The first author used the Kernel Tuner and Network Builder to experiment with hand-crafting weights as a means of creating line drawings. In addition, the author explored various calibrations of the generative algorithm to produce artistic output. The following sections provide an overview of case studies for each tool and a short description of the artistic process utilizing the tools.

Findings & Discussion

Kernel Tuner

As a specific example, the Kernel Tuner was used to produce a set of weights for a line end detector for the drawing system (Fig. 2). After about an hour of experimentation, a set of eight kernels of five pixels across was chosen (Fig. 2a). Fewer and smaller kernels minimized the number of calculations performed during convolution and thus allowed the algorithm to operate more efficiently. Since the detector did not have to be perfect, these kernels were an attractive balance between speed and accuracy. It is important to note that the eight kernels produced were not a one-size fits all solution for detecting line ends in images. Instead, they were a solution for a particular creative project with a certain type of image.

A distinct advantage of the Kernel Tuner was the visibility into the effects of different options. The explanatory visuals provided insights into the relative balance between kernels. Initially, our intuition was that these convolutional kernels would give each pixel a name such as “upwards facing line end” or “left facing right angle.” Closely watching the activations and interacting with the system demonstrated how difficult it was to disentangle signals. As more types and rotations of kernels were added many kernels were activated at any given pixel. This suggested that the internal abstractions in a network are better thought of in terms of adjectives on continuous scales rather than one-hot vectors of labels matching nouns. A patch of pixels cannot simply be labeled as a “corner.” It is “corner”-like, but also “vertical line”-like and “horizontal line”-like. The Kernel Tuner supported exploration, produced useful kernels, and yielded insights into the internal representations of neural networks.

Network Builder

The Network Builder was used to create flexible detectors of increasingly complex visual concepts including boxes (Fig. 3), houses, and bottles (Fig. 4). One of the first networks designed with the Network Builder was for robustly detecting different-sized boxes. To start, the network was designed using positive weights corresponding to vertical and horizontal lines. This attempt produced extra lines and lacked corners (Fig. 3a). Various patterns of construction were explored, such as overlap between parts, sizes of kernels, the depth of negative margins, and the relative magnitude of weights (Fig. 3b and c). After hours of experimentation, the proper combination of positive connections to lines and corners, as well as negative margins, yielded flexibly shaped boxes with a single continuous line and no extra artifacts (Fig. 3d).

Artistic Output

Next, we explored the production of artistic outputs using these tools. The generative system (using the bottle-detecting network as a reward function) could be run on user inputs to evaluate the success of the weights. If the samples do not match expectations, the generative algorithm is run again until it reproduces an aspect that is not intended (such as premature stopping).
input as seen in Fig. 4 or on a blank canvas as seen in Fig. 5. Fig. 5 demonstrates a more complex workflow. Multiple versions of the generative system maximizing different parts of the bottle-detecting network (starting from the lower layers and moving up) were run. The artistic process involved writing small programs for running the generative algorithm as well as tweaking thresholds and parameters within the algorithm (such as the ratio of drawing to erasing or when to halt). Fig. 5 also demonstrates the further step of printing out and hand painting the result.

Our goal with this conceptual work was to leverage the system’s strength, which is its ability to robustly and dynamically respond to input, and to see if direct manipulation of network weights could be used to make intentional output. The findings demonstrate that the system is able to respond to human or random input and to transform it into rough, but recognizable, shapes.

**Conclusion**

This paper has proposed and evaluated two tools for hand-crafting networks for art-making. The purpose of these tools was to facilitate manipulating network weights by providing information at the point of action. This research demonstrated the possibility of discovering insights and making intentional, granular decisions through direct engagement, yet the successes were relatively minor. The findings indicated that crafting more complex visual concepts within networks would likely require data-driven methods. To address this, future work could explore targeted training while maintaining the interactive visualizations inside the networks. One possibility would be to manipulate weights through synthetic datasets coupled with rich labeling at multiple levels of abstraction. Operating on the interior of neural networks presents challenges, but understanding the abstract encodings contained within them is an intriguing reward.

**Acknowledgments**

We are grateful for input from Golan Levin and members of the Computational Design Laboratory at CMU.

**References**


