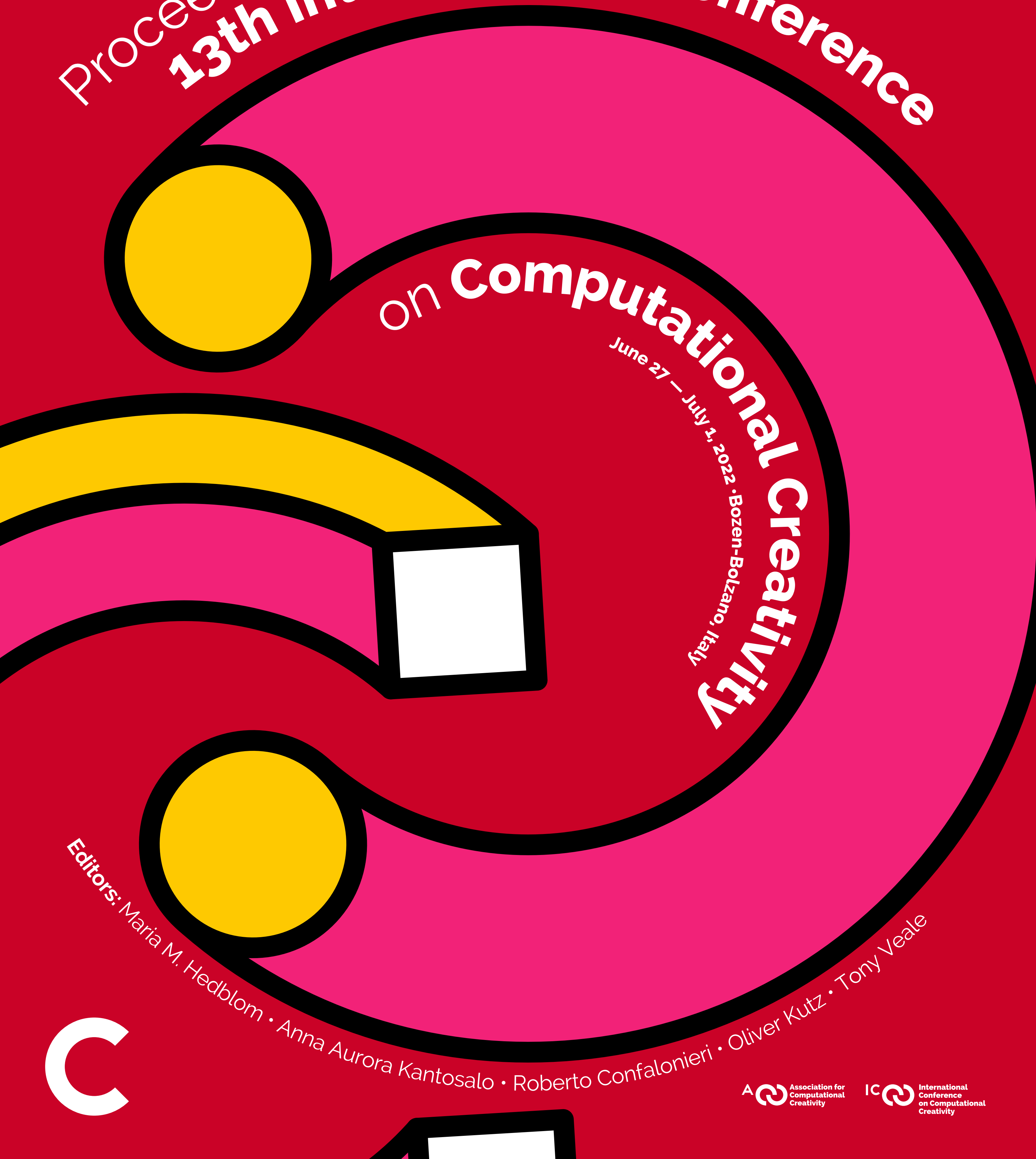
Proceedings of the 13th International Conference on Computational Creativity

June 27 — July 1, 2022 · Bozen-Bolzano, Italy



Editors: Maria M. Hedblom · Anna Aurora Kantosalo · Roberto Confalonieri · Oliver Kutz · Tony Veale

Association for Computational Creativity

International Conference on Computational Creativity

Proceedings of the Thirteenth International Conference on Computational Creativity

ICCC'22
Bozen-Bolzano, Italy — 27 June - 1 July

Maria M. Hedblom, Anna Aurora Kantosalo,
Roberto Confalonieri, Oliver Kutz and Tony Veale (Editors)

Published by the Association for Computational Creativity
(ACC)

Association for
Computational
Creativity

Free University of Bozen-Bolzano
Italy

http://computationalcreativity.net/iccc22/

First published 2022

TITLE: Proceedings of the 13th International Conference on Computational Creativity

EDITORS: Maria M. Hedblom, Anna Aurora Kantosalo, Roberto Confalonieri, Oliver Kutz and Tony Veale

ISBN: 978-989-54160-4-2

Published by the Association for Computational Creativity (ACC)

ISBN 978-989-54160-4-2

9 789895 416042

Preface

Each year, researchers and practitioners interested in computational creativity gather for the International Conference on Computational Creativity, ICCC. From June 27 to July 1st, 2022, the 13th edition of the conference was held in Bozen-Bolzano, tucked in between the green mountains of northern Italy. After two years of pandemic lockdown with conferences taking place in the cloud, this marked a welcomed return to physical meetings of the community. With both its feet on the ground, ICCC'22 also ensured that it had its head in the clouds and the whole conference was streamed to offer a remote presentation format to the participants that were unable to travel.

The local organisational team from the Free University of Bozen-Bolzano ensured that the conference rooms were in order and, together with the session chairs, that the conference was running smoothly. Throughout the conference week, the catering team served coffee and Italian pastries to keep the energy high among the participants, and to facilitate an as sustainable conference setting as possible, a predominantly vegetarian lunch was served daily.

For the scientific program of the 13th International Conference on Computational Creativity, the conference received 126 submissions: 57 to the long paper track and 69 to the short paper track. Out of these, 55 were accepted for presentation and inclusion in the proceedings: 22 as long papers and 33 as short papers.

Displaying an internationally diverse community, the accepted papers represent work of researchers and practitioners from 48 academic institutions, ten enterprises and one private practitioner, in turn spread over 17 countries: Australia, Canada, Finland, France, Germany, India, Ireland, Italy, Japan, Luxembourg, Malta, the Netherlands, Saudi Arabia, Slovenia, Spain, the United Kingdom and the United States of America.

We extend our gratitude to all authors who submitted their manuscripts to this conference, and congratulate the authors of accepted papers for their excellent contributions presented in this volume. Particularly outstanding were the following papers, which won the best paper awards in three different categories:

Best Paper Award: Patrick Chieppe, Penny Sweetser and Eryn Newman: "Bayesian Modelling of the Well-Made Surprise"

Best Short Paper: Kana Maruyama and Michael Spranger: "Interpretable Relational Representations for Food Ingredient Recommendation Systems"

Best Student Paper: Marvin Zammit, Antonios Liapis and Georgios Yannakakis: "Seeding Diversity into AI Art"

The proceedings is the result of a highly selective peer-reviewing procedure conducted by the Program Committee. Invited for their expertise in computational creativity and closely related topics, the members in the committee helped ensure the quality and appropriate fit of the papers selected for inclusion in the conference. We note that as the period since the previous conference was only nine months, both submitting authors and the PC had to work under an unusually narrow time frame. In addition, we acknowledge the difficult conditions for this year's PC as yet another wave of Covid-19 broke out during the reviewing period. In light of this, we extend our warmest thanks to those who pushed through and despite these challenges gave of their valuable time.

For the long paper track, each paper was reviewed and discussed by at least three PC members. Based on these reviews and personal expertise, a senior PC member wrote a meta-review that included a recommendation for acceptance or rejection. The final decisions were made by the Program Chairs in relation to the reviews, the meta-review and the recommendations of the senior PC members. Similarly, for the short paper track, each submission received three independent reviews. However, since there was less time between the deadline for short papers and the conference, there was no time for a meta-review period for the short paper track. To assist the Program Chairs in making the final decisions, the PC members had instead been instructed to be as consise in their reviews as possible by either leaning towards an acceptance or a rejection recommendation.

Another thing worth mentioning is that this year we wanted to pay extra attention to inclusiveness and transparency. This manifested in two ways: The selection of the Program Committee and the addition of two new sections in the accepted papers.

First, when selecting the Program Committee, we paid extra attention to diversity. To this end, we looked for new talent among the authors of previous editions of the conference and especially invited members from universities outside Europe and North America. While doing this, we also paid attention to the gender balance of the committee and did our best to balance the scale. For reviewing of the short papers, we also invited new members of research areas closely related to computational creativity. The motivation for this was a wish to broaden the skillset of the PC, expand the community and promote the research area of computational creativity to researchers that might be interested in participating in upcoming editions of the conference. We are happy that so many answered our call and were willing to participate!

Second, to increase transparency and openness of the proceedings, accepted papers were required to include two new sections: *Acknowledgements* and *Author Contributions*. The purposes of these sections are to recognise the influence of individuals, sponsors and institutions behind the research, and to in more detail describe the contribution each author had to the papers. We believe that such transparency is threefold in its benefits. First, that it supports good research practices of the rich variety of scientific fields uniting under computational creativity research. Second, that it helps our authors in getting an accurate level of acknowledgement for their work. Thirdly, that it offers our readers assistance in finding the correct researchers to contact for particular questions about the papers.

The scientific program of ICCC'22 held, in addition to the paper presentations, book release presentations, a demo session and four highly diverse keynote speakers that also joined in a panel debate.

Book release presentations: To support the community, three book release presentations were held at the conference:

- Mike Sharples and Rafael Perez y Perez: "Story Machines: How Computers Have Become Creative Writers"
- Nada Lavrač, Vid Podpečan and Marko Robnik-Šikonja: "Representation Learning: Propositionalization and Embeddings"
- Tony Veale: "Your Wit Is My Command: Building AIs with a Sense of Humor"

Demo session: All authors who had requested to demonstrate their systems along with their paper presentations were able to participate in the demo session. The different systems were showcased in 5-min flash talks followed by show-and-tell at individual monitors where the audience could walk around and engage with the systems. The demo session was a fast-paced, exciting display of computational creativity systems ranging from artistic domains to more practical applications in information science.

Keynote speakers: With a special emphasis for having representatives from industry, research and the arts, as well as a large span of traditional computational creativity areas; visual arts, text generation, multi-media and computational humour, our four keynotes were:

- Aaron Hertzmann: "Can computers create art?"
- Allison Parrish: "Material Paratexts of Computer-Generated Literature"
- Ellen Pearlman: "Biometrics, AI, Embodiment, Performative Practices and the Speculative Future"
- Oliviero Stock: "A Catchy Title"

The 13th edition of ICCC spanned a week of academic activities. Preceding the three days of the main conference, ICCC'22 also consisted of two days of workshops, tutorials and a doctoral consortium.

Workshops: Highly appreciated by the audience, three particular foci were offered in the workshops: The cultural heritage of Internet art, the role of embodiment in creativity, and what computational creativity could offer in terms of a therapeutic tool. The workshops were:

- "Double View But Single Vision? Tracing Artistic Internet Browsers" by Barbara Filser, Inge Hinterwaldner, Daniela Hönigsberg, and Konstantin Mitrokhov
- "The Role of Embodiment in the Perception of Human and Artificial Creativity" by Laura Herman and Caterina Moruzzi
- "Therapeutic Computational Creativity & the Third Hand" by Maya Ackerman and Alison Pease

Tutorials: In comparison to the workshops, the tutorial offered a more hands-on educational environment to spread novel insights to the community. Topic-wise very diverse, the two tutorials were:

- "Methods of interacting with computational systems for live artistic performance" by Kory Mathewson, Thomas Winters, Piotr Mirowski, and Boyd Branch
- "Quantum Computing for Computational Creativity" by Alain Lioret

The doctoral consortium: To ensure that the computational creativity community grows the next generation of researchers were invited to participate in the DC. 21 students from varying backgrounds and at different stages of their studies participated and out of these 16 received a scholarship. The event consisted of five-minute flash talks to the audience at ICCC that were followed by group discussions with two assigned senior researchers. To participate, each student had to submit a summary of their research that had been evaluated by the DC chairs and the ICCC'22 organisation. Separate proceedings for the DC contributions is available.

Social program: Further, to ensure that the computational creativity community remains a friendly group of researchers that happily invites new members, ICCC'22 took special care to arrange for a social program with great networking opportunities. Social highlights included:

- A conference reception at Museon, Bozen-Bolzano's modern art museum
- An afternoon at Firmian Messner Mountain Museum
- A conference dinner at Castle Maretsch
- An electro-acoustic music concert at the Conservatory Claudio Monteverdi: Cello & live electronics, audiovisuals and multichannel acousmatic music by Prof. Gustavo Delgado's electronic music class and special guest Prof. Nicola Baroni.

As demonstrated in these proceedings, the International Conference on Computational Creativity boasts a multidisciplinary community. Each year, we invite authors from various fields to participate in the conference and as a community, we have tight connections to representatives from different branches within the arts. As the research field of computational creativity has matured, it has spread its wings and become a subject of interest in various other conferences and events as well. However, ICCC remains an inclusive conference that warmly welcome different technical solutions and methods over all areas of creativity research. The 13th edition of the conference particularly excels in representing this domain-independent core of the discipline itself in that many of the papers in these proceedings tackle a key concept beyond a specific methodological discipline. This remains a unique feature of the ICCC conference and emphasises its importance as a publication venue for computational creativity regardless of the increasing popularity at other venues.

Acknowledgements The ICCC'22 organisational team expresses its gratitude to its sponsors: The Free University of Bozen-Bolzano (unibz), the Association for Computational Creativity (ACC), the European Association for Artificial Intelligence (EURAI), Adobe Research, and the Artificial Intelligence Journal (AIJ). Specifically, we would like to thank unibz and ACC for the general organisational support, EURAI for sponsoring the participation of our invited keynote speaker Oliviero Stock, Adobe for sponsoring the Best Short Paper award, and AIJ for sponsoring the Best Paper and Best Student Paper awards.

We thank Prof. Gustavo Delgado and his students, as well as special guest Prof. Nicola Baroni, for the impressive electroacoustic music concert held at Bolzano's Music Conservatory Claudio Monteverdi.[1]

We acknowledge the guidance of the ACC steering committee and the CC Task Force and we extend our thanks to those who made ICCC'22 possible: the organisational team, including all the chairs, the program committee and additional reviewers, as well as our student helpers. Finally, we thank you, the reader of these proceedings, for your interest and for picking up the ideas and work presented in this volume and turning it into the next wave of valuable contributions of future research!

[1] See the details here https://computationalcreativity.net/iccc22/music-concert/ and watch it here https://www.youtube.com/watch?v=f1uWrPjT8wU.

Organisational Team

General Chairs

Oliver Kutz - Free University of Bozen-Bolzano
Tony Veale - University College Dublin

Program Chairs

Maria M. Hedblom - Jönköping University
Anna Aurora Kantosalo - University of Helsinki

Local Chair

Roberto Confalonieri - Free University of Bozen-Bolzano

Workshops and Tutorials Chair

Diarmuid O'Donoghue - Maynooth University

Doctoral Consortium Chairs

Roberto Confalonieri - Free University of Bozen-Bolzano
Oliver Kutz - Free University of Bozen-Bolzano

Demo Chair

João Miguel Cunha - University of Coimbra

Publicity Chair

Maria Teresa Llano Rodriguez - Monash University

Web Chair

Guendalina Righetti - Free University of Bozen-Bolzano

Program Committee

Senior PC Members

Maya Ackerman - Santa Clara University
Kat Agres - National University of Singapore
Oliver Bown - Interactive Media Lab, UNSW
Daniel Brown - University of Waterloo
F. Amílcar Cardoso - University of Coimbra
Pietro Galliani - Free University of Bozen-Bolzano
Kazjon Grace - The University of Sydney
Ashok Goel - Georgia Institute of Technology
Anna Jordanous - University of Kent
Sarah Harmon - Bowdoin College
Raquel Hervás - Universidad Complutense de Madrid
Nada Lavrač - Jozef Stefan Institute
Carlos León - Universidad Complutense de Madrid
Mary Lou Maher - University of North Carolina Charlotte
Rafael Pérez Y Pérez - Universidad Autonoma Metropolitana
Senja Pollak - Jozef Stefan Institute
Rob Saunders - Leiden University
Hannu Toivonen - University of Helsinki
Dan Ventura - Brigham Young University
Geraint Wiggins - Vrije Universiteit Brussel / Queen Mary University of London

PC Members

Taisuke Akimoto - Kyushu Institute of Technology
Mihailo Antovcić - University of Niš
Gabriella Barros - Modl.ai
John Bateman - University of Bremen
Tarek Richard Besold - TU Eindhoven
Daniel Beßler - University of Bremen
Paul Bodily - Idaho State University
Marcel Bollmann - Jönköping University
Stefano Borgo - Laboratory for Applied Ontology, ISTC-CNR
Chun-yien Chang - National Yang Ming Chiao Tung University
Ying-ping Chen - National Yang Ming Chiao Tung University

Liuqing Chen - Zhejiang University
Simon Colton - Queen Mary University of London
João Correia - University of Coimbra
João Miguel Cunha - University of Coimbra
Stefano De Giorgis - Alma Mater University of Bologna
Shlomo Dubnov - University of California in San Diego
Mohamed Elhoseiny - KAUST
Milena Fisher - The Creativity Post
Björn Gambäck - Norwegian University of Science and Technology
Fabrício Góes - University of Leicester
Andrés Gómez de Silva Garza - Instituto Tecnológico Autónomo de México
Hugo Gonçalo Oliveira - University of Coimbra
Christian Guckelsberger - Aalto University
Ivan Guerrero - UNAM
Matthew Guzdial - University of Alberta
Jer Hayes - Accenture
Colin Johnson - University of Nottingham
Maximos Kaliakatsos-Papakostas - Aristotle University of Thessaloniki
Janin Koch - Inria Paris-Saclay
Antonios Liapis - University of Malta
Antonio Lieto - University of Turin
Heather Ligler - Pennsylvania State University
Simo Linkola - University of Helsinki
Phil Lopes - Universidade Lusófona
Róisín Loughran - Dundalk Institute of Technology
Nuno Lourenço - University of Coimbra
Penousal Machado - University of Coimbra
Pedro Martins - University of Coimbra
Najma Mathema - Brigham Young University
Jon McCormack - Monash University
David Meredith - Aalborg University
Tomi Männistö - University of Helsinki
María Navarro - University of Salamanca
Santiago Negrete-Yankelevich - Universidad Autónoma Metropolitana
Diarmuid O'Donoghue - Maynooth University
Allison Parrish - New York University
Philippe Pasquier - Simon Fraser University
Rafael Peñaloza - University of Milano-Bicocca
Giovanni Pilato - ICAR-CNR

H. Sofia Pinto - Instituto Superior Tecnico
Mihai Pomarlan - Institute for Artificial Intelligence, University of Bremen
Guendalina Righetti - Free University of Bolzano
Maria Riveiro - Jönköping University
Ana Rodrigues - University of Coimbra
Melissa Roemmele - Language Weaver (RWS Group)
Sebastian Rudolph - TU Dresden
Emilio M. Sanfilippo - Laboratory for Applied Ontology, ISTC-CNR
Marco Schorlemmer - Artificial Intelligence Research Institute (IIIA), CSIC
Brad Spendlove - Brigham Young University
Niclas Ståhl - Jönköping University
Tapio Takala - Aalto University
Alan Tapscott - Universitat Pompeu Fabra
Kıvanç Tatar - Chalmers University of Technology
Prashanth Thattai Ravikumar - De Montfort University
Martin Thiering - Technische University Berlin
Tiago Torrent - Federal University of Juiz de Fora
Nicolas Troquard - Free University of Bozen-Bolzano
Nikolaos Tsiogkas - KU Leuven
Alessandro Valitutti - Phedes Lab
Doug Van Nort - York University
Olga Vechtomova - University of Waterloo
Florian Westphal - Jönköping University
Lonce Wyse - National University of Singapore
Pinar Yanardag - Bogazici University
Georgios N. Yannakakis - University of Malta
Haizi Yu - University of Chicago
Martin Žnidaršič - Jožef Stefan Institute

Contents

1. Generating narratives

Casual Poetry Creators: A Design Pattern and Internal Evaluation Measures

Michele Boggia♡ **Sardana Ivanova**♠ **Simo Linkola**♠ **Hannu Toivonen**♠ **Anna Kantosalo**♠
♡ Department of Digital Humanities ♠ Department of Computer Science
University of Helsinki, Finland
{first.last}@helsinki.fi

Abstract

We explore the concept of Casual Poetry Creators with the aim of making poetry writing fun and entertaining for the user. We present a simple co-creative interaction design pattern based on constructing poems line by line, suggesting the user a set of line candidates at each step. We also propose objective measures by which a Casual Poetry Creator can evaluate and choose which line candidates to show to the user and sketch out a plan to evaluate the measures and pattern with users.

Introduction

Writing poetry is a creative act. Poets do it for various reasons—to communicate a feeling or a viewpoint, for self-expression or for therapeutic reasons, for instance. In this paper, we address people who are not versed with poetry but who could nevertheless have joy from writing it—given that they had access to easy-to-use tools that make the threshold to try out poetry writing very low.

We explore the concept of *Casual Poetry Creators*, systems that use a simple interaction pattern with the aim of making poetry writing fun and entertaining for novices. Casual Creators, a term coined by Compton and Mateas (2015), refers to a class of co-creative tools characterized by playfulness and the lack of task-focus. The main goal for Casual Poetry Creator systems, then, is not to generate great poetry, but rather to help the user feel the joy of creativity.

We contribute two elements to Casual Poetry Creators:

First, the defining element of our Casual Poetry Creator is a simple interaction pattern where poems are generated line by line, with the user in control over which lines are used in the poem. Specific design advice on casual creators has been published in the form of design patterns (Compton and Mateas 2015; Compton 2019; Petrovskaya, Deterding, and Colton 2020) and case studies of designing suitable parameter spaces for casual creation e.g. in the domains of games (Colton et al. 2018) and visual arts (Colton et al. 2020). Several simple, interactive poetry generators have been proposed, too. However, as far as we know, this is the first time the task is considered within the casual creators framework. The actual poetry generation method is outside the scope of this paper; instead, we present methods in a separate paper (Boggia et al. 2022). Casual Poetry Creators can be implemented with different generation methods, e.g., sequence-to-sequence linguistic models to generate lines. We hope that our work encourages researchers to contribute novel Casual Poetry Creators based on their models.

Second, we define objective evaluation measures for assessing candidate lines for poetry. These measures have several applications: (a) with these measures, a Casual Poetry Creator can internally evaluate its line candidates, so as to provide an appropriate set to the user; (b) when designing a new Casual Poetry Creator, the measures can be used to assess the suitability of different poetry generators for casual creation; (c) during the building of a Casual Poetry Creator, the measures can help fine-tuning linguistic models for this particular purpose.

This paper is structured as follows. In the next section, we briefly review background in casual creators and interactive poetry writing. We then introduce the interaction pattern for Casual Poetry Creators. Next, we give definitions of objective measures that can be used to implement Casual Poetry Creators. We wrap this paper up with concluding remarks. In a parallel paper (Boggia et al. 2022), we give poetry generation algorithms that are suitable for Casual Poetry Creators, we describe implementations of the objective measures, and we give empirical results.

Background

Casual creators

The concept of Casual Creators (Compton and Mateas 2015) gives a name for an old phenomenon covering both physical tools as well as software characterized by assistance, automation and limiting the domain space of possible creative outputs to support novice creators (Compton 2019, p. 20).

Compton and Mateas (2015) define a casual creator as an interactive system that encourages fast, confident, and pleasurable exploration of a possibility space, resulting in the creation or discovery of surprising new artifacts that bring feelings of pride, ownership, and creativity to the users that make them.

Casual creators offer an interesting platform for computational creativity developers to develop applications for use in the real world. Examples of casual creation emerge in physical toys, as part of other, more complex software, such as character creation tools within games, and as tools or games are re-used for casual creation instead of their original pur-

Proceedings of the 13th International Conference on Computational Creativity (ICCC'22)
ISBN: 978-989-54160-4-2

pose (Compton 2019, p. 6, 11, 14). Dedicated applications conforming with casual creation are also readily available on commercial app platforms, such as the Apple App Store (Petrovskaya, Deterding, and Colton 2020), further speaking to their role as a widely available form of pass-time for novice creators. In addition, Casual creators offer opportunities to create well-being for their users (Compton 2019, p. 3), making them a significant area to improve the outreach of computational creativity research.

The goal of our Casual Poetry Creator interaction pattern and the metrics are the same as with any casual creator systems: they focus on the users' enjoyment of the creative process itself above productivity and scaffold the creative process by enabling the rapid and fluent exploration of a restricted creative space (Compton 2019, p. 6–7).

Interactive Poetry Generators

Poetry generation is a popular research topic in computational creativity and numerous methods have been proposed in the literature. A review of different techniques to generate poetry is out of the scope of this paper, however, and we refer the interested reader to Gonçalo Oliveira (2017).

Interactive poetry generation where the software acts as an intelligent or creative partner has also been addressed by several scholars. The following are representative examples of interactive poetry writing systems.

The Poetry Machine (Kantosalo et al. 2014; Kantosalo, Toivanen, and Toivonen 2015) uses a fridge magnet metaphor for interaction. The system starts with a sample poem generated by the system, and then the user can move words or lines around, write more text, or ask the system to suggest new words or lines.

Hafez (Ghazvininejad et al. 2017) produces sonnets based on given words and eight style parameters tuned by the user. The interaction model is based on the user adjusting the parameters and asking the system to regenerate the poem according to the new parameters.

Machine in the Loop (Clark et al. 2018) is an approach used for writing text line by line in other fields of creative writing such as stories and poetry. At every iteration, the system suggests a line which the user may then edit.

Co-PoeTryMe (Oliveira et al. 2019) includes user interface functions similar to the Poetry Machine, and additionally it allows constraints to be specified for new words (e.g. rhyme) and offers an extensive editing functionality.

We focus on poetry writing applications using a very simple exploration method for the creative conceptual space of poetry. The space is initialized based on user-given input keywords, and subsequent direction of poetry generation takes place through the one-touch and mutant-shopping interaction patterns for casual creators (Compton and Mateas 2015; Petrovskaya, Deterding, and Colton 2020).

The Casual Poetry Creator design pattern that we describe in the next section is simpler than in any of the above systems. Our aim is to make the use of Casual Poetry Creators as simple as possible: no parameters to tune, no different user interface functions to choose from—not necessarily even an option to edit lines produced by the system, which removes the need for a keyboard when writing poetry with a Casual Poetry Creator. For the input of keywords that function as seeds for line generation, different keyboard-free alternatives can be considered such as pointing at words in a document, or offering the user a set of random words from which to select the seeds.

A Design Pattern for Casual Poetry Creators

We consider a simple model for co-creative poetry generation. The poetry generator produces poetry one line at a time, in simple interaction with the user:

1. Before the generation of the first line, the user may give a couple of keywords;
2. Candidates for the first line are produced with the keywords as inspiration; if no keywords were provided, first line candidates are produced based on random keywords sampled from a dictionary;
3. The user selects one of the candidate lines suggested by the system;
4. Candidates for the next lines are produced based on the previous lines (and potentially the keywords);
5. The poem is constructed iteratively and incrementally by going back to step 3; the user may decide to stop the generation of new lines at any time, in which case the final poem is printed by the system.

This design allows very simple user interaction. At any time, the system provides a handful of candidate lines from which the user chooses one, and the system then generates candidates for the next line based on the previous selections of the user.

The generation of candidate lines should satisfy three criteria: (1) each candidate line is related to the previous lines in the poem, or to the possible keywords in the case of the first line; (2) each candidate line is poetic; and (3) the candidates for the nth line are diverse.

In this paper we focus on this simple interaction pattern for two reasons. First, we believe this makes a good Casual Creator, with extreme simplicity for the user but still a wide space of possible poems (assuming the poetry generator is successful in the three goals listed above). Second, the simple interaction pattern can be easily reused with other poetry generation methods, as well as in Casual Creators in other domains where incremental creation of artefacts is natural.

In Listing 1, we present an example implementation of the Casual Poetry Generator pattern as a command line interface.[1] The example shows the keyword input and a few iterations of the creation process including the line candidate examples and the current lines in the poem.

The line-by-line candidate generation model can be seen as an instance of several design patterns of Casual Creators (Compton and Mateas 2015). It gives instant feedback in the form of new lines based on the user's selection; it produces many candidates and overlays them; it avoids the blank canvas problem by writing lines for the user; it offers limited actions to encourage exploration; and it's mutant shopping

[1] A Python implementation is available at https://github.com/bmichele/poetry_generation.

Listing 1: Example outputs of the command line implementation of a casual poetry generator.

```
GIVE KEYWORDS: nature summer

LINE CANDIDATES:
0 Nature, in the summer's heat,
1 Nature, in summer's sunshine-bright,
2 Nature, like Summer's with her own decrees,
3 And summer's charms, by Nature given,
4 The summer's nature and the summer's love,
5 Nature, in summer time, is still the same,
6 The summer's breath of nature blows,
7 And Nature, like a Summer's flowery mist,

PLEASE CHOOSE CANDIDATE
(integer in [0,1,2,3,4,5,6,7], -1 to stop.)

[...]

CURRENT POEM STATE:
 The summer's breath of Nature blows,
 Across the fields and through the trees,
 Its fragrance, like the breath of May,

LINE CANDIDATES:
0 A rose-leaf in the garden-bough,
1 The breath of flowers blown in the breeze.
2 A rose-bush in the midst of May,
3 A rose-bush in the garden-bough,
4 A rose-leaf in the garden breeze.
5 A rose-bush in the morning breeze.

PLEASE CHOOSE CANDIDATE
(integer in [0,1,2,3,4,5], -1 to stop.)
```

in the sense that it offers alternative lines ready to be picked by the user (however instead of changing the whole artifact, our pattern focuses on additive iteration). Saving and sharing is trivial since the approach only operates on and produces text.

Perhaps the closest parallels within existing casual creator patterns are 'Limiting Actions to Encourage Exploration' and 'Mutant Shopping' (Compton and Mateas 2015). Casual Poetry Creators could be implemented in a mobile interface with a 'One-touch creativity' pattern, which uses only one type of gesture for the interaction (Petrovskaya, Deterding, and Colton 2020). The basic interaction offered by the pattern could of course be extended, by for example allowing the user to edit the lines. Such an extended pattern begins to resemble user interactions with existing co-creative poetry writing systems, such as the Poetry Machine (Kantosalo et al. 2014) or Co-PoeTryMe (Oliveira et al. 2019).

Internal Evaluation Measures

We propose four evaluation measures to assess poetry lines produced by Casual Poetry Creators: semantic coherence, topic coherence, tautology, and diversity. In this paper, we do not aim to measure how poetical lines are.

These measures can be utilised both (1) by the designer of the system during the system development to assess the feasibility of the generation methods and (2) by the system itself during its execution time to make informed decisions about which set of generated line candidates to show to the user. The evaluation measures are based on metrics and other measures previously proposed in the literature.

Preliminaries For two vectors, the *cosine similarity* is defined as the cosine of the angle θ between the vectors.

For two sets of tokens S_1, S_2 (lines consisting of words), *token similarity* $\mathrm{sim}\left(S_1, S_2\right)$ is defined based on their overlap as

$$\mathrm{sim}(S_1, S_2) = \frac{2|S_1 \cap S_2|}{|S_1| + |S_2|}. \tag{1}$$

Semantic Coherence Candidate lines offered by a Casual Poetry Creator should usually be semantically coherent with the poem written so far. For this purpose, we define the *n-Semantic Coherence* to measure the semantic similarity of a candidate line to the n previous lines. This measure can be used by a Casual Poetry Creator to decide which lines to show the user. For instance, the measure could be used to select a set of candidates mostly consisting of lines coherent with the previous ones, but also include a few less coherent ones to allow for surprises and turns in the poem.

The n-Semantic Coherence of a candidate verse for the ith line of the poem is defined as follows. We consider the n previous lines, i.e., lines $i - n$ to $i - 1$, transform them to a vector representation and compute its cosine similarity with a vector representation of the candidate line.

More specifically, we tokenize each line, remove stopwords, and compute the centroid of the word vectors obtained for each token of the n previous line from the Word2vec model (Mikolov et al. 2013a; 2013b). The n-semantic coherence of the candidate is then the cosine similarity between this vector and the vector obtained from the candidate line by following the same procedure (tokenization, stopword removal, computation of centroid by averaging word vectors).

The idea is that the two vectors are semantic encodings of the last lines of the poem and of the candidate line, respectively, and that their cosine similarity captures the degree of their semantic similarity. Line candidates introducing new subjects into the poem will have lower semantic coherence.

Topic Coherence Candidate lines suggested by a Casual Poetry Creator should usually be related to the keywords given by the user (if any). We define *Topic Coherence* of a candidate line as its semantic similarity with the keywords. A Casual Poetry Creator can use the topic coherence in ways analogical to semantic coherence, e.g., to ensure that the set of candidate lines contains both topic coherent and potentially surprising lines.

Technically, the topic coherence of a candidate line is defined as the cosine similarity between the centroid of (the word embeddings of) the line and the centroid of (the word embeddings obtained from) the user-given keywords.

The idea is to extend the concept of semantic coherence defined above and offer means to measure the topic drift of candidate lines from the initial keywords. Candidates characterized by lower scores, when compared with the input keywords, would look more surprising but potentially incoherent to the user. High values, in turn, imply lower surprise and higher coherence.

Tautology Many sequence-to-sequence language models are prone to produce unnecessarily repetitive lines, or *Tautology*, and safe-guarding against them can be needed. (For instance, in our implementation (Boggia et al. 2022) we use mBART, which is pre-trained on denoising tasks (Liu et al. 2020). If fine-tuning is not successful, the model will tend to repeat the same verse(s) over and over again.) A measure of tautology allows a Casual Poetry Creator to filter our repetitive lines, if needed.

For a candidate line, we define tautology as the number of tokens that are shared between the candidate and the previous line of the poem, normalized by the total number of tokens in the two verses. We can express this measure using token similarity simply as $\mathrm{sim}(S_i, S_{i-1})$, where S_i and S_{i-1} are the sets of words obtained from the candidate and the previous poem, respectively.

Diversity A Casual Poetry Creator should produce a diverse set of candidate lines at each generation step. This ensures that the user has a real choice and is more likely to feel ownership and pride of the resulting poem. We define the *Diversity* of a *set* of lines by the amount of words shared between them. Usually, a Casual Poetry Creator would try to maximize the diversity in the candidate lines it offers to the user.

To measure the diversity of a set of lines, we utilise token similarity $\mathrm{sim}(S_1, S_2)$ between two lines, where S_1 and S_2 are the set of words extracted from the lines. The diversity is computed as the average *dissimilarity* between the lines in the line set, where dissimilarity between two word sets S_1 and S_2 is $1 - \mathrm{sim}(S_1, S_2)$. That is, for a set of poem lines, we first extract the words from them to obtain a set of word sets $\mathbf{S} = (S_1, \ldots, S_n)$, and then compute diversity $\mathrm{div}(\mathbf{S})$ in a following manner:

$$\mathrm{div}(\mathbf{S}) = \frac{n(n-1)}{2} \sum_{i=0}^{n-1} \sum_{j=i+1}^{n} \big(1 - \mathrm{sim}(S_i, S_j)\big). \quad (2)$$

Empirical Validation and Application

In a parallel paper (Boggia et al. 2022), we empirically validate that the semantic coherence and diversity metrics measure what they are supposed to, and argue that topic coherence and tautology will also behave favorably. We also apply these measures on an actual poetry generation method and report on our empirical findings.

Planned External Evaluations

We have implemented a Casual Poetry Creator as a command line interface running on a local instance, using the poetry generation method of Boggia et al. (2022). Basic evaluation with end users is already possible with this interface, but we intend to implement it as a web-based tool for easier access. Offering the system as a web service will also allow easier systematic evaluation of co-creative experiences of users. It would be interesting to investigate the relationship between the internal evaluation metrics and users' co-creative experiences with the system, offering further insight into the beneficial use of these metrics in systems aiming for Casual Poetry Creation.

Conclusion

We presented a simple interaction design pattern to facilitate the creation of Casual Poetry Creators. The pattern is based on line-by-line generation and selection of poem contents, and is well suited for human-computer co-creation.

The Casual Poetry Creator design pattern only allows very simple user interaction. The user starts the interaction by providing a small set of keywords. Candidates for the first line are then produced with these keywords as inspiration, and the user selects one of the lines. After that, candidates for the next lines are produced based on the previous lines. The poem is constructed iteratively and incrementally in this manner, until user decides to stop.

The interaction is highly limited on purpose. The goal is to make the threshold for poetry writing as low as possible by keeping the interface simple. This follows the Casual Creator philosophy: the aim of the system is to help the novice user feel joy of creativity, not to accomplish a task.

A successful Casual Poetry Creator has the ability to produce and select suitable sets of candidate lines. We argue that in Casual Poetry Creators, good candidate lines should be coherent with the preceding poem as well as poetic; additionally, the set of candidates should have diversity. While poetry generation methods are outside the scope of this paper, we proposed evaluation measures that can be used as internal filters by different Casual Poetry Creators. The proposed metrics measure the coherence, diversity, and repetition in lines of poetry. Whatever method is used to generate alternative lines, these measures can be used to control what kind of candidate sets are offered to the user.

This paper is a first conceptual step towards Casual Poetry Creators. In a parallel paper (Boggia et al. 2022), we propose matching poetry generation methods, and validate and apply internal evaluation measures. The next step is an evaluation of the concept and of the measures with actual users. Do the users get joy of creativity, and how do various factors—as potentially indicated by the measures proposed—affect how much fun it is? On the technical side, we plan to explore measures related to the concept of poeticalness, e.g. by measuring poetic devices such as rhyming and using machine learning, either as part of learning to generate new lines of poetry, or as a separate evaluation step.

Author Contributions

The concept of Casual Poetry Creators was developed through discussions between all authors. AK acted as a co-creativity expert and wrote the introduction, background and the description of the casual creation pattern together with HT. MB, SI, SL and HT formulated the internal evaluation measures and wrote the corresponding sections and conclusions. All authors contributed to the writing of the manuscript iteratively.

Acknowledgments

MB has been partially supported by the ERC Consolidator Grant FoTran (agreement № 771113) and by the grant 825153 (EMBEDDIA) under the European Union's Horizon 2020 research and innovation programme.

SI has been funded by the European Union's Horizon 2020 research and innovation program under grants 825153 (EMBEDDIA) and 770299 (NewsEye). SL and AK have been funded by Academy of Finland (Grant #328729).

References

Boggia, M.; Ivanova, S.; Linkola, S.; Kantosalo, A.; and Toivonen, H. 2022. One line at a time — generation and internal evaluation of interactive poetry. In *Proceedings of the 13th International Conference on Computational Creativity*. ACC.

Clark, E.; Ross, A. S.; Tan, C.; Ji, Y.; and Smith, N. A. 2018. Creative writing with a machine in the loop: Case studies on slogans and stories. In *Proceedings of the 23rd International Conference on Intelligent User Interfaces*, IUI '18, 329–340. New York, NY, USA: ACM.

Colton, S.; Nelson, M.; Powley, E.; Gaudl, S.; Saunders, R.; Perez Ferrer, B.; Ivey, P.; and Cook, M. 2018. A parameter-space design methodology for casual creators. In *Proceedings of the Ninth International Conference on Computational Creativity*, 264–271. ACC.

Colton, S.; McCormack, J.; Berns, S.; Petrovskaya, E.; and Cook, M. 2020. Adapting and enhancing evolutionary art for casual creation. In *Artificial Intelligence in Music, Sound, Art and Design*, 17–34. Springer.

Compton, K., and Mateas, M. 2015. Casual creators. In *Proceedings of the Sixth International Conference on Computational Creativity*, 228–235. Brigham Young University.

Compton, K. E. 2019. *Casual Creators: Defining a Genre of Autotelic Creativity Support Systems.* UC Santa Cruz.

Ghazvininejad, M.; Shi, X.; Priyadarshi, J.; and Knight, K. 2017. Hafez: an interactive poetry generation system. In *Proceedings of The 55th Annual Meeting of the Association for Computational Linguistics, System Demonstrations*, 43–48. ACL.

Gonçalo Oliveira, H. 2017. A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. In *Proceedings of the 10th International Conference on Natural Language Generation*, 11–20. Santiago de Compostela, Spain: ACL.

Kantosalo, A.; Toivanen, J. M.; Xiao, P.; and Toivonen, H. 2014. From isolation to involvement: Adapting machine creativity software to support human-computer cocreation. In *Proceedings of the Fifth International Conference on Computational Creativity*, 1–7. Jožef Stefan Institute.

Kantosalo, A.; Toivanen, J. M.; and Toivonen, H. 2015. Interaction evaluation for human-computer co-creativity: A case study. In *Proceedings of the Sixth International Conference on Computational Creativity*, 276–283. Brigham Young University.

Liu, Y.; Gu, J.; Goyal, N.; Li, X.; Edunov, S.; Ghazvininejad, M.; Lewis, M.; and Zettlemoyer, L. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics* 8:726–742.

Mikolov, T.; Chen, K.; Corrado, G. S.; and Dean, J. 2013a. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations (Workshop Presentation)*.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*. NIPS.

Oliveira, H. G.; Mendes, T.; Boavida, A.; Nakamura, A.; and Ackerman, M. 2019. Co-PoeTryMe: interactive poetry generation. *Cognitive Systems Research* 54:199–216.

Petrovskaya, E.; Deterding, C. S.; and Colton, S. 2020. Casual creators in the wild: A typology of commercial generative creativity support tools. In *Proceedings of the Eleventh International Conference on Computational Creativity*. ACC.

One Line at a Time — Generation and Internal Evaluation of Interactive Poetry

Michele Boggia[♡] **Sardana Ivanova**[♠] **Simo Linkola**[♠] **Anna Kantosalo**[♠] **Hannu Toivonen**[♠]
[♡] Department of Digital Humanities [♠] Department of Computer Science
University of Helsinki, Finland
{first.last}@helsinki.fi

Abstract

We present methods that produce poetry one line at a time, in a manner that allows simple interaction in human-computer co-creative poetry writing. The methods are based on fine-tuning sequence-to-sequence neural models, in our case mBART. We also consider several internal evaluation measures by which an interactive system can assess and filter the lines it suggests to the user. These measures concern the coherence, tautology, and diversity of the candidate lines. We empirically validate two of them and apply three on the mBART-based poetry generation methods. The results suggest that fine-tuning a pre-trained sequence-to-sequence model is a feasible approach, and that the internal evaluation measures help select suitable models as well as suitable lines.

Introduction

We propose methods that use sequence-to-sequence neural models to generate poetry one line at a time. We use them to implement a simple interaction pattern where the system iteratively produces a set of line candidates from which the user picks one, with the aim of making poetry writing easy and entertaining for novices.[1]

We also consider four objective evaluation measures to assess candidate lines especially in interactive poetry generation. While we suggest these measures elsewhere (Boggia et al. 2022) we have not evaluated or applied them before. In this paper, we empirically validate them and show how to use the measures in practical applications.

Poetry generation is a popular research topic in computational creativity and numerous methods have been proposed in the literature (Gonçalo Oliveira 2017). Interactive poetry generation where the software acts as an ìntelligent or creative partner has also been addressed by several scholars (Kantosalo et al. 2014; Kantosalo, Toivanen, and Toivonen 2015; Ghazvininejad et al. 2017; Oliveira et al. 2017; 2019; Clark et al. 2018).

Our poetry generator produces poetry one line at a time, based on the previous lines of the poem or, in the case of the first line, based on keywords given by the user. This approach supports different user interaction patterns where suggestions for continuation are requested from the system. In this paper, we assume the following simple interaction pattern of Casual Poetry Creation (Boggia et al. 2022).

At any time, the system provides a handful of candidate lines from which the user chooses one. The system then iteratively generates candidates for the next line based on the user's previous selections. Candidates for the first line are generated from keywords inserted by the user. The system can be easily adapted to allow more complex interaction patterns, such as allowing the user to edit the system outputs.

The generation of candidate lines in such an interactive setting should satisfy three criteria: (1) each candidate should be related to the previous lines in the poem, or to the keywords in the case of the first line; (2) each candidate line should be poetic; and (3) the set of candidates for the nth line should be diverse. In the next section, we present poetry generation methods that address points 1 and 2; in the following section, we use the internal measures to address points 1 and 3.

Poetry Generation with mBART

To generate poems line by line we leverage mBART (Liu et al. 2020), a denoising autoencoder pre-trained on monolingual corpora in several languages. In a nutshell, the model takes a *source sequence* (e.g., a partially written poem) and produces a *target sequence* (the next line of the poem).

Starting from the same base model, we fine-tune (i) a model to generate candidates for the first poem line from the input keywords, and (ii) a model that generates candidates for additional poem lines. We will refer to these neural models as *first-line* and *next-line model* respectively.

The fine-tuning datasets for our models are constructed from the *Gutenberg Poetry Corpus*[2], a corpus of approximately 3M poem lines in English language extracted from Project Gutenberg[3].

First-Line Model

In our interaction pattern the first line of a poem is generated based on a small, unordered set of input keywords provided

[1] A Python implementation of the system is available at https://github.com/bmichele/poetry_generation.

[2] https://github.com/aparrish/gutenberg-poetry-corpus

[3] https://gutenberg.org

by the user.

In the fine-tuning step, we use the first lines of stanzas in the corpus as target texts. Since we do not have keywords for the poems or stanzas we obtain keyword proxies from the first lines by selecting two or more random content tokens among the nouns, adjectives and verbs on the line. The source text for each fine-tuning example is obtained by shuffling and concatenating the tokens.

Next-Line Models

At every iteration, after a line is selected by the user, the system should provide a set of candidates for the next line of the poem. Since there is no clear prescription on the best way to generate additional lines for a poem, we consider several options for fine-tuning mBART. We start by considering the previous line only, and progressively move towards more complex strategies. This allows us to compare candidate lines generated with different degrees of context. In general, we expect to obtain more surprising outcomes when the generation is based on a lower amount of textual input.

The first model we consider, *Next-Line Single*, is fine-tuned as follows: we iterate over all the lines in the corpus and build examples taking a poem line and its subsequent line as source and target sequence, respectively. We do not expect this model to produce lines that remain coherent with the user keywords after the first few iterations.

To get more coherent verses, we train two additional models: "Next-Line Multi" and "Next-Line Keywords". The *Next-Line Multi* approach fine-tunes mBART by using up to three consecutive poem lines as source sequence; the target is the verse following the input lines. The *Next-Line Keywords* approach increases coherence by conditioning next-line generation on the keywords obtained from the user.

The fine-tuning data is similar to the *Next-Line Single* model; for the Next-Line Keywords model we additionally prepend to the source sequence a pair of words related to the target sequence. To obtain them we first compute the average word vector of the target sequence tokens using Word2vec (Mikolov et al. 2013a; 2013b). We then retrieve the ten closest words in the Word2vec model by cosine similarity, and randomly sample two of them.

The fine-tuning strategies described above rely on the assumption that the base model, when fine-tuned over poem lines, will naturally learn to produce poetic line candidates. However, there is no control over how this is learned by the models and it will be influenced by the data that is present in the original corpus.

In the *Next-Line Rhyme* case we fine-tune a model that tries to generate, given a word and a line, a new poem line rhyming with the given word. Giving the word separately allows to produce lines rhyming with earlier lines, not just the previous one.

The fine-tuning data is similar to the data used for the *Next-Line Single* model, but we prepend to the source sequence a word rhyming with the target sequence; we use the CMU Pronouncing Dictionary[4] to look up rhymes. When no rhymes are found, we discard the pair. If multiple rhymes are available, we randomly sample up to four examples.

We fine-tune all the models for 10 epochs using batches of 64 examples over 4 GPUs. Due to the different preprocessing steps taken to build the fine-tuning data, the size of the datasets are slightly different for each model, resulting in a number of fine-tuning steps that is between 90k and 95k. We save model checkpoints every 15k steps.

Decoding Strategy

A good set of candidate lines is diverse, offering the user a real choice. An autoencoder such as mBART produces output sequences stochastically, so several candidates can be generated from the same model.

We generate candidates by sampling multiple sequences from the probabilities predicted by the fine-tuned models. In this way, the output sequences do not follow a distribution of high probability next tokens but are less predictable and can surprise the user (Holtzman et al. 2020). The randomness — and diversity — of the output can be controlled by the *temperature* parameter: values larger than one will increase the likelihood of low probability tokens.

Internal Evaluation Measures for Poetry

We consider four evaluation measures to assess the lines produced by the above poetry generation models. These measures can be utilised both (1) by the designer of the system during the system development to assess the feasibility of the generation methods and (2) by the system itself during its execution time to make informed decisions about which set of generated line candidates to show to the user.

Measures

We give a brief overview of the measures here. See our parallel paper (Boggia et al. 2022) for details.

We define the *n-Semantic Coherence* of a candidate for the ith line of the poem as follows. We consider the n previous lines, i.e., lines $i-n$ to $i-1$, transform them to a vector representation and compute its cosine similarity with a vector representation of the candidate line. Both vector representations are obtained computing the centroid of word vectors from the Word2Vec model. The idea is that the two vectors encode the semantic of the last lines of the poem and the candidate, respectively, and that their cosine similarity captures the degree of semantic similarity.

We define *Topic Coherence* of a candidate line as the cosine similarity between the vector representation of the line and the average of the word embeddings of the keywords used to generate the first poem line. The idea is to extend the concept of semantic coherence defined above and make it suitable to our interaction pattern in order to control the topic drift of each candidate from the initial keywords.

For a candidate line, we define *Tautology* as the number of tokens that are shared between the candidate and the previous line of the poem, normalized by the total number of tokens in the two lines. We consider this metric as our sequence-to-sequence models are based on mBART, which is pre-trained on denoising tasks and can be prone to copy

[4] `https://github.com/cmusphinx/cmudict`

the input sequence if not fine-tuned properly. Semantic coherence and tautology are likely to be correlated since both measure a similarity between consecutive lines. The former aims, however, at comparing meanings, while the latter compares words. An incoherent poem will display low semantic coherence scores; a high tautology value will indicate a repetitive poem.

We want our candidate poem lines on each generation step to be sufficiently different from each other, to give the user a true choice. We define the *diversity* of a set of lines as the average dissimilarity between the lines in the line set, where dissimilarity between two word sets is the additive inverse of the normalized word overlap between the sets.

Validation Datasets

We next validate the evaluation measures introduced above, showing that they do indeed measure what they are supposed to measure. Given the lack of suitable labelled data that could be used directly for this purpose, we resort to resampling lines from real poems. We can then compare the values of the measures on original poems against those obtained for random pseudo-poems.

To validate the measures, we use a dataset of poems from *Kaggle*, containing 6321 poems.[5] While the Poetry Corpus introduced in the previous section is suitable for training neural models, it is not optimal to validate the metrics of this section as it contains noisy data and there is no separation between poems.

Starting from the original poems, here referred to as *Real Poems*, we prepare two additional poem datasets. We randomly shuffle lines *within* each of the real poems and obtain the *Shuffled Poems* dataset. We then build the *Mixed Poems* dataset by shuffling lines *between* poems. To ensure equal poem line counts in the mixed poems we compute the poem lengths from each real poem and construct poems with the same number of lines by sampling (without replacement) from all the available poem verses in the corpus.

Validation of the Measures

The four metrics described above can be divided into two classes based on their implementation. First, semantic and topic coherence make use of word vectors to map textual inputs in a semantic space and compare them by computing the cosine similarity. Second, tautology and diversity are based on word overlap and rely solely on token sets. In this paper we validate the semantic coherence and diversity measures and argue that topic coherence and tautology will display a similar behaviour.

To validate the n-semantic coherence we consider the datasets with real, shuffled and mixed poems. Our hypothesis is that we should observe decreased semantic coherence scores after shuffling poem lines within a poem, and much lower values when considering pseudo-poems obtained by combining lines from random poems.

For each dataset we compute the n-semantic coherence of all poem lines (excluding first lines) with the n previous lines up to the first one or $n = 10$. Finally, we average the semantic coherence values for each order n (Figure 1). The average values of n-semantic coherence scores are systematically smaller when shuffling poem lines, with the lowest average values obtained when poems are composed of random lines.

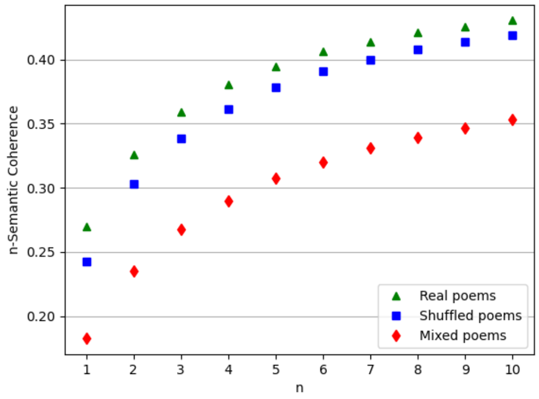


Figure 1: n-semantic coherence scores of Real Poems, Shuffled Poems and Mixed Poems as a function of n, the number of previous lines considered.

Temperature	Diversity	Diversity without stopwords
1	0.1406	0.1039
2	0.4014	0.3429
3	0.4788	0.4232
4	0.5174	0.4434
5	0.5377	0.4701

Table 1: Average diversity scores of sets of candidate lines as a function of temperature, a generation-time parameter affecting diversity.

To inspect how the diversity measure behaves when computed over different sets of poem line candidates, we rely on synthetic data produced by our first-line model. We construct 100 random keyword pairs by combining nouns and verbs sampled from a list of common English words. For each keyword pair, we use different temperature parameter of the model to generate multiple batches of ten candidates each. Candidates generated with higher temperatures are more diverse by construction.

As expected, the diversity increases as a function of the temperature (Table 1). This is true for the full lines (middle column) as well as when stopwords have been removed before computation of diversity (right column). This validates that the diversity measure does catch differences in the generated lines.

Analysis of the mBART-Based Methods with Internal Evaluation Measures

We now apply the internal evaluation measures on the mBART-based poetry generation methods. With this brief

[5] https://www.kaggle.com/michaelarman/poemsdataset

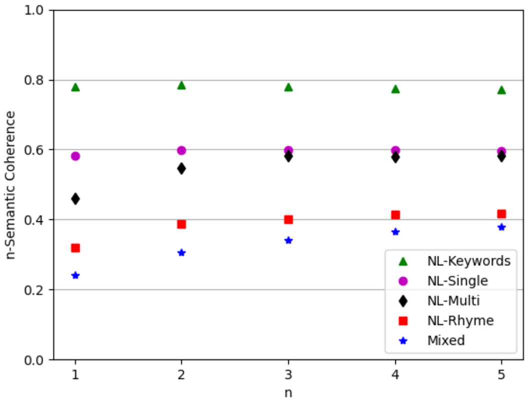


Figure 2: n-semantic coherence scores for poems generated by different Next-Line models, as a function of n, the number of previous lines considered.

Model	Tautology ↓	Diversity ↑
NL-Keywords	0.206	0.184
NL-Single	0.145	0.431
NL-Multi	0.096	0.459
NL-Rhyme	0.046	0.352
Mixed	0.045	0.841

Table 2: Average tautology and diversity scores for line candidates generated using different Next-Line models. For tautology, smaller values are better; for diversity, larger.

example we aim to shed light on the specific generation methods, as well as to illustrate how the evaluation measures can be used to assess generation methods. In this experiment, we compare the four flavours of the Next-Line model previously described, using the evaluation measures for semantic coherence, tautology and diversity.

In order to test the generation methods without user interaction, we generate poems automatically by sampling the first poem line from a dataset of real poems and then selecting a random candidate line at each generation step. We stop the generation process after ten candidate selections. To collect data for diversity assessment, we also log the line candidates proposed by the generator at each iteration. We use the above procedure to obtain 100 poems and 1000 sets of candidates with each of the four Next-Line models.

As a baseline, we fine-tune a model over random poem lines from the Gutenberg Poetry Corpus both as source and target sequences. This model, called *Mixed*, gives a lower bound for the coherence of poems.

We report the n-semantic coherence scores of the resulting poems in Figure 2, and their tautology and diversity scores in Table 2. Based on these results we can make several observations about the models, such as the next two.

The *NL-Keywords* model, introduced to avoid topic drift, effectively improves the coherence of the poems (Figure 2), but the price to pay is that poems become repetitive and have low diversity (Table 2). A qualitative inspection of the generated poems confirms this finding. For instance, this poem was obtained with keywords “bury” and “dead”:

And let the dead folk bury their dead,
But let the dead men bury their dead,
Let the dead men bury their dead,
Let the living men bury their living,
Let the dead folk sleep.
. . . .

The *NL-Multi* model, on the other hand, produced relatively interesting poems even without human interaction:

Which tries, and counter-stands the shock,
Of time and chance;
And, having learn’d to bear the yoke,
To bear the yoke must learn to love,
And follow Truth, and all that’s above.
The ways that lead to Heaven’s high throne,
Are long and hard to tell;
But this way leads to God alone,
And that way leads to Hell.

The success of the ML-Multi model in this respect is no surprise: it obtained both high semantic coherence scores as well as a high diversity score.

A different but important application for the internal measures is the optimization of the set of candidates towards a desired feature. For instance, assume that the system fails to satisfy the user because of a lack of diversity in the candidates. The sequence-to-sequence models could then be used to generate a larger number of potential candidates (which in our setup is computationally inexpensive), and they could then be narrowed down to a final set of candidates while maximising their diversity.

Conclusion

We gave several variants of fine-tuned mBART-models for line-by-line poetry generation. One variant produces opening lines from keywords, while other models produce lines to continue a partially written poem. The models consider varying contextual information: one or more previous lines, user-given keywords, or rhyme. The methods are designed in a manner that should allow relatively easy adaptations to different genres, corpora, and even languages.

We empirically validated internal evaluation measures of lines of poetry. We showed that the proposed measures of coherence and diversity correlate with ground truth.

Finally, we applied three evaluation measures on generation methods that continue an incomplete poem. The results indicate trade-offs between the methods. The NL-Multi method that uses several lines as a context seems to strike a good balance.

The choice to work line-by-line, both in generation and in internal evaluation of poetry, stems from the desire to support Casual Poetry Creation (Boggia et al. 2022) and to make co-creative poetry writing as easy as possible. The next step is an evaluation of the approach with actual users.

Author Contributions

MB, SI and HT conceived the poetry generation methods. MB, SI, SL and HT selected and defined the internal evaluation measures. MB and SI processed the raw data for the experiments. SI fine-tuned the first-line model, MB the next-line models. MB and SL implemented and validated the metrics. MB was in charge of writing the technical part; everybody contributed to the writing of the other sections.

Acknowledgments

MB has been partially supported by the ERC Consolidator Grant FoTran (agreement № 771113) and by the grant 825153 (EMBEDDIA) under the European Union's Horizon 2020 research and innovation programme.

SI has been funded by the European Union's Horizon 2020 research and innovation program under grants 825153 (EMBEDDIA) and 770299 (NewsEye). SL and AK have been funded by Academy of Finland (Grant #328729).

References

Boggia, M.; Ivanova, S.; Linkola, S.; Toivonen, H.; and Kantosalo, A. 2022. Casual Poetry Creators: A design pattern and internal evaluation measures. In *Proceedings of the 13th International Conference on Computational Creativity*. ACC.

Clark, E.; Ross, A. S.; Tan, C.; Ji, Y.; and Smith, N. A. 2018. Creative writing with a machine in the loop: Case studies on slogans and stories. In *Proceedings of the 23rd International Conference on Intelligent User Interfaces*, IUI '18, 329–340. New York, NY, USA: ACM.

Ghazvininejad, M.; Shi, X.; Priyadarshi, J.; and Knight, K. 2017. Hafez: an interactive poetry generation system. In *Proceedings of The 55th Annual Meeting of the Association for Computational Linguistics, System Demonstrations*, 43–48. ACL.

Gonçalo Oliveira, H. 2017. A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. In *Proceedings of the 10th International Conference on Natural Language Generation*, 11–20. Santiago de Compostela, Spain: ACL.

Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; and Choi, Y. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations (Poster Presentation)*.

Kantosalo, A.; Toivanen, J. M.; Xiao, P.; and Toivonen, H. 2014. From isolation to involvement: Adapting machine creativity software to support human-computer co-creation. In *Proceedings of the Fifth International Conference on Computational Creativity*, 1–7. Jožef Stefan Institute.

Kantosalo, A.; Toivanen, J. M.; and Toivonen, H. 2015. Interaction evaluation for human-computer co-creativity: A case study. In *Proceedings of the Sixth International Conference on Computational Creativity*, 276–283. Brigham Young University.

Liu, Y.; Gu, J.; Goyal, N.; Li, X.; Edunov, S.; Ghazvininejad, M.; Lewis, M.; and Zettlemoyer, L. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics* 8:726–742.

Mikolov, T.; Chen, K.; Corrado, G. S.; and Dean, J. 2013a. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations (Workshop Presentation)*.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*. NIPS.

Oliveira, H. G.; Hervás, R.; Díaz, A.; and Gervás, P. 2017. Multilingual extension and evaluation of a poetry generator. *Natural Language Engineering* 23(6):929–967.

Oliveira, H. G.; Mendes, T.; Boavida, A.; Nakamura, A.; and Ackerman, M. 2019. Co-PoeTryMe: interactive poetry generation. *Cognitive Systems Research* 54:199–216.

Modern French Poetry Generation with RoBERTa and GPT-2

Mika Hämäläinen[1,2], Khalid Alnajjar[1,2] and Thierry Poibeau[2]
[1]University of Helsinki, Finland
[2]École Normale Supérieure-PSL and CNRS and Université Sorbonne nouvelle, Paris, France
firstname.lastname@{helsinki.fi[1] or ens.psl.eu[2]}

Abstract

We present a novel neural model for modern poetry generation in French. The model consists of two pretrained neural models that are fine-tuned for the poem generation task. The encoder of the model is a RoBERTa based one while the decoder is based on GPT-2. This way the model can benefit from the superior natural language understanding performance of RoBERTa and the good natural language generation performance of GPT-2. Our evaluation shows that the model can create French poetry successfully. On a 5 point scale, the lowest score of 3.57 was given by human judges to *typicality* and *emotionality* of the output poetry while the best score of 3.79 was given to *understandability*.

Introduction

Poem generation is a challenging creative natural language generation task. As a form of art, it has undergone several changes in the history. Classical poetry incorporates typically meter and rhyme as their function was to help people recall poems, especially when poetic tradition was still mostly oral rather than written.

In the modern era, the role of the poetry has changed from an art form that has to follow a fixed structure that defines its meter and rhyming such as iamb, haiku or anapest. Modern poetry is more concerned about creating something new by breaking any strict structural rules and by continuously questioning what poetry is, what it can be and what it should be (see Kantokorpi, Lyytikäinen, and Viikari 1990).

In the field of poem generation, meter is a feature that is very often considered in generated poetry (Colton, Goodwin, and Veale 2012; Lau et al. 2018; Hämäläinen and Alnajjar 2019b; Zugarini, Melacci, and Maggini 2019; Lewis, Zugarini, and Alonso 2021). By incorporating meter, people can be more forgiving when evaluating the output of the system as it is known that people are ready to interpret more into the content of the output of a computationally creative system if the form is correct (Veale 2016). In other words, a poem that looks like a poem, as in that it follows a certain meter, must be a poem. A truly competent computational poet should be capable of generating something that is recognizable as a poem even if its output was modern free-form poetry.

In this paper, we explore the topic of modern poetry generation in French. We fine-tune a novel encoder-decoder architecture which consists of a RoBERTa (Liu et al. 2019) based model as the encoder and a GPT-2 (Radford et al. 2019) based model as the decoder. Because RoBERTa is very good at natural language understanding tasks but poor at generation tasks and GPT-2 is good at generation, but bad at understanding, it makes sense to incorporate both of the models. The task of RoBERTa is to encode the input (i.e. to understand poetry) and the task of GPT-2 is to decode the output (i.e. to generate poetry).

Related work

Poem generation has sparked a lot of interest in the past as we can see in a recent survey on the field (Gonçalo Oliveira 2017). There is also some work on generating French poetry in particular (Van de Cruys 2019; Poibeau et al. 2020). In this section, we will provide a quick overview on some of the related work.

Poetry has been generated by using rules. This can, for instance, be done by modeling certain poetic features to find suitable verses (Lamb and Brown 2019), by writing different rules to generate verses with different poetic features (Hämäläinen 2018) or by using templates (Colton, Goodwin, and Veale 2012).

Genetic algorithms have been popular in poem generation as well (Manurung, Ritchie, and Thompson 2012; Yang et al. 2016; 2021). In many of these approaches, the generation itself is not governed by rules, but rules are used to assess the quality of each generated poem. These approaches can be seen as a step away from fully relying on rules in the generative process.

Recently, different neural approaches have become common place in poetry generation such as LSTM models (Zhang and Zhang 2020; Atassi and El Azami 2022) and GPT-2 (Hu and Sun 2020; Beheitt and Hmida 2022). Our approach follows this line of work. How we set ourselves apart from the existing models is that we propose a novel poem generator model that can benefit from both RoBERTa and GPT-2 instead of relying only on one model. In addition, we condition the beam search on rhyme instead of solely relying on the output of a vanilla model.

Verse	Input	Output	Output in English
1	amant galamment escrime dame	Vainement, paladin des dames, tu t'escrimes	In vain, paladin of the ladies, you fight me with a sword
2	Vainement, paladin des dames, tu t'escrimes	À crier que j'ai fait, ici, peu galamment.	Shouting what I have done, here, little gallantly.
3	À crier que j'ai fait, ici, peu galamment.	si de Laure Pétrarque avait été l'amant,	if Pétrarque had been the lover of Laure
4	si de Laure Pétrarque avait été l'amant,	Qui sait ce qu'il eût dit de Laure dans ses Rimes?	Who knows what he said about Laure in his Rhymes?

Table 1: Example of the training data for one poem

Data

As machine learning requires data, we need a poem corpus. For this reason, we crawl all the French poems that are available on Wikisource[1]. The poems are not free of noise as some of the poems include verses in Greek alphabets, multiple different types of quotation marks, hyphens and spaces of different lengths etc. We clean the data from all of these inconsistencies by manually inspecting odd characters and either by replacing them (e.g. only one type of a hyphen) or removing them (e.g. Greek letters). The corpus contains 7553 poems. In addition, we use the French sonnet corpus introduced by Poibeau et al. 2020. This corpus has 1039 sonnets.

Because these poems and sonnets are of different lengths, we split all of them into stanzas. From this point on, we treat a stanza as a poem so that all poems in our corpus are of a similar length. This gives us altogether 25,215 French poems and sonnets. For the purposes of our models, we do not make a distinction between poems and sonnets.

Poem generator

In this section, we describe our poem generation model. The model follows an encoder-decoder architecture where the encoder is a RoBERTa model and the decoder is a GPT-2 model. Rather than training these models from scratch, we use pretrained language models and fine-tune them for the task of poem generation using a transfer learning approach. We chose a RoBERTa-based model as the encoder given their great ability in capturing contextual semantics. GPT-2 is well-known for modeling a language; hence, making an optimal decoder for text-generation tasks.

First we have to pick the suitable pretrained models. As we use Transformers library (Wolf et al. 2020), we select our models from their repository. The current state-of-the-art French RoBERTa model is CamemBERT[2] (Martin et al. 2020) which is based on the RoBERTa (Liu et al. 2019) architecture and trained on the large OSCAR corpus (Abadji et al. 2022) in French. We use CamemBERT as our encoder.

As for the selection of the GPT-2 model, there were several alternatives. By trying the models out, we could see that all of them except for Belgian GPT-2[3] (Louis 2020) predicted rather poor output. The model was trained on a variety of genres (such as news, Wikipedia, novels, European parliament text etc.) on a relatively big, around 60 GB, corpus. For this reason, we opted for Belgian GPT-2 as our decoder model.

We use Spacy[4] (Honnibal et al. 2020) to extract up to 4 keywords from each poem in the corpus. We train our encoder-decoder architecture for sequence to sequence generation, where it predicts the next verse in a poem given a previous verse. In the absence of a previous verse, we train the model to predict the first verse of a poem from the up to 4 keywords extracted from the poem. An example of input and output in the training data for one poem can be seen in Table 1. The first input consists of the keywords *amant* (lover), *galamment* (gallantly), *escrime* (fencing) and *dame* (lady), which are used to predict the first verse of the poem.

The poem corpus is split randomly to 80% for training and 20% for validation. The model is trained for 10 epochs. We use the Adam algorithm (Kingma and Ba 2014) with decoupled weight decay regularization (Loshchilov and Hutter 2017) and learning rate of 5e-05 to optimize the parameters of the model, with cross entropy loss as the loss function to reduce the difference between gold standard token and predicted tokens.

Rhyming is taken into account during the generation phase. The model is requested to generate a sequence between 4 to 20 tokens, with a length penalty of 1.0 using a greedy approach. At each step of generating the output sequence (i.e., when predicting the next token), we use the model to predict the top 10 possible tokens instead of just one highest scoring output. We then sort these candidate tokens based on their probabilities and rhyming scores. The rhyming score is calculated by counting the number of tokens in the output that rhyme (full rhyme, consonance or assonance) with the input (i.e., the previous verse and any subsequent words generated during the run).

Because it is not easy to know whether two French words rhyme or not based on the orthography (similarly to English), we use eSpeak-ng[5] to produce an IPA (international phonetic alphabet) representation for each token in the model's vocabulary. IPA alphabets are designed to represent how words are pronounced by writing out the actual phonemes. We use a simple set of rules to compare the IPA strings of two tokens with each other to determine whether they rhyme or not.

In practice, we first ensure that both of the IPA strings are equally long, if this is not the case, we remove characters from the beginning of the longer string until the IPA strings are equally long. If the strings are identical, no rhyme is considered, because a word does not make a good rhyme with itself. For full rhyme, the two IPA strings rhyme if they are identical from the first vowel onward. For assonance, we replace all consonants with a placeholder character *C*, if the

[1] https://fr.wikisource.org/wiki/Catégorie:Poèmes

[2] https://huggingface.co/camembert-base

[3] https://huggingface.co/antoiloui/belgpt2

[4] The *fr_core_news_sm* model

[5] https://github.com/espeak-ng/espeak-ng

D'un beau travail, d'une bonne pose,	From a beautiful work, from a good pose.
De la paix, de la beauté.	From the peace, from the beauty
Que je plains la beauté	Oh, I lament the beauty
De la femme, qui m'inspire	Of the woman, who inspires me
C'est ici que s'éveille le soleil,	It is here where the sun wakes
C'est ici que repose le grand créateur,	It is here where the great creator rests
Dont la ruine, hélas! se renouvelle	Whose ruin, alas! renews itself
De l'Enfant du Progrès	From the Child of Progress
C'est un des mois les plus beaux de l'année,	It is one of the most beautiful months of the year,
C'est le printemps, c'est l'été, c'est	It is the spring, it is the summer, it is
Le ciel où mon printemps se joue.	The sky where my spring plays.
À mon jardin qui s'effondrit.	In my garden that collapses

Table 2: Examples of generated poetry and their translations.

IPA strings are identical, i.e. they share the same vowels in the same positions, they are considered to have assonance rhyme. For consonance, we do the same as with assonance, but by replacing all vowels with a placeholder *V*.

Results and evaluation

For evaluation purposes, we generate 20 different poems consisting of 4 verses each. For each poem, we use a set of four randomly selected keywords among all the keywords extracted from the poem corpus. None of the keyword combinations is identical to what the model saw during the training. We generate the poems similarly to the example shown in Table 1. This means, that the keywords were used to generate the first verse, which was then used to generate the second verse and so on.

Some of the generated poems and their translations can be seen in Table 2. As we can see, the generated output is cohesive and quite grammatical. We can, however, see that sometimes the verb conjugation might be wrong such as in the case of *effondrit* which is a non-existing inflectional form of *effondrer* (to collapse). Also, the model has a tendency of starting every verse with a capital letter even if it was a continuation to the sentence started in the previous verse.

We conduct a crowd-sourced evaluation on Appen[6]. We set French as a language requirement for the crowd-workers so that we know that they actually speak French and are able to assess French poetry. Each poem is evaluated by 20 different crowd-workers. An individual worker can evaluate all 20 different poems or just some of them, in which case the remaining unevaluated poems are shown to a different crowd-worker. An individual crowd-worker cannot evaluate the same poem multiple times.

For evaluation, we use the same parameters as used by several authors for evaluating poetry (Toivanen et al. 2012; Hämäläinen and Alnajjar 2019a; Shihadeh and Ackerman 2020): (1) *The poem is typical* (2) *The poem is understandable* (3) *The poem is grammatical* (4) *The poem evokes imagination* (5) *The poem evokes emotions* (6) *I like the poem.* These statements are evaluated in a 5 point Likert scale, where 1 represents the worst and 5 the best grade.

	Q1	Q2	Q3	Q4	Q5	Q6
Avg	3.57	3.79	3.77	3.65	3.57	3.77
STD	0.88	0.84	0.81	0.79	0.88	0.77

Table 3: The evaluation results and standard deviation

The results can be seen in Table 3. All in all, the results are good and show that the system can generate poetry successfully. The lowest scores were obtained for *typicality* and *emotionality*. The highest score was given to *understandability*. In the future, more robust human evaluation methods need to be applied to understand why these parameters scored high and low (Hämäläinen and Alnajjar 2021a; 2021b).

Conclusions

In this paper, we have presented a novel approach to French poem generation. We have presented an architecture that consists of RoBERTa and GPT-2 models that are fine-tuned on a poem corpus. In addition, we have modeled rhyme as a part of the prediction pipeline of the model.

The results obtained in human evaluation are promising and they indicate that the model performs well in the task it was designed to do. In order to make the evaluation results more transparent, we have released them in full on Zenodo[7] together with the generated poems that were used in the evaluation.

Pretrained neural language models have been proven to be useful in poem generation. In the future, it would be interesting to study them in a multilingual setting, where a pretrained multilingual model is fine-tuned to generate poetry using the corpora of some languages other than the desired target language.

Author Contributions

The first two authors contributed to the work presented in this paper equally. The third author was involved in planning the methods and writing the paper.

[6] https://appen.com/

[7] https://zenodo.org/record/6558357

Acknowledgments

This work was partially financed by the Society of Swedish Literature in Finland with funding from Enhancing Conversational AI with Computational Creativity. This work was conducted during a mobility period supported by Nokia Foundation under grant number 20220193. This work was supported in part by the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR19-P3IA-0001 (PRAIRIE 3IA Institute). The work was also supported by the CNRS funded International Research Network Cyclades (Corpora and Computational Linguistics for Digital Humanities).

References

Abadji, J.; Ortiz Suarez, P.; Romary, L.; and Sagot, B. 2022. Towards a Cleaner Document-Oriented Multilingual Crawled Corpus. *arXiv e-prints* arXiv:2201.06642.

Atassi, A., and El Azami, I. 2022. Comparison and generation of a poem in arabic language using the lstm, bilstm and gru. *Journal of Management Information & Decision Sciences* 25.

Beheitt, M. E. G., and Hmida, M. B. H. 2022. Automatic arabic poem generation with gpt-2. In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART,*, 366–374. INSTICC.

Colton, S.; Goodwin, J.; and Veale, T. 2012. Full-face poetry generation. In *ICCC*, 95–102.

Gonçalo Oliveira, H. 2017. A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. In *Proceedings of the 10th International Conference on Natural Language Generation*, 11–20. Santiago de Compostela, Spain: Association for Computational Linguistics.

Hämäläinen, M., and Alnajjar, K. 2019a. Generating modern poetry automatically in Finnish. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 5999–6004. Hong Kong, China: Association for Computational Linguistics.

Hämäläinen, M., and Alnajjar, K. 2019b. Let's face it. finnish poetry generation with aesthetics and framing. In *Proceedings of the 12th International Conference on Natural Language Generation*, 290–300.

Hämäläinen, M., and Alnajjar, K. 2021a. The great misalignment problem in human evaluation of NLP methods. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, 69–74. Online: Association for Computational Linguistics.

Hämäläinen, M., and Alnajjar, K. 2021b. Human evaluation of creative NLG systems: An interdisciplinary survey on recent papers. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, 84–95. Online: Association for Computational Linguistics.

Hämäläinen, M. 2018. Harnessing nlg to create finnish poetry automatically. In *Proceedings of the ninth international conference on computational creativity*. Association for Computational Creativity (ACC).

Honnibal, M.; Montani, I.; Van Landeghem, S.; and Boyd, A. 2020. spacy: Industrial-strength natural language processing in python, 2020. *URL https://doi.org/10.5281/zenodo* 1212303(6).

Hu, J., and Sun, M. 2020. Generating major types of chinese classical poetry in a uniformed framework. *arXiv preprint arXiv:2003.11528*.

Kantokorpi, M.; Lyytikäinen, P.; and Viikari, A. 1990. *Runousopin perusteet*. Gaudeamus.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. In *arXiv*.

Lamb, C., and Brown, D. G. 2019. TwitSong 3.0: towards semantic revisions in computational poetry. In *Proceedings of the Tenth International Conference on Computational Creativity*, 212–219.

Lau, J. H.; Cohn, T.; Baldwin, T.; Brooke, J.; and Hammond, A. 2018. Deep-speare: A joint neural model of poetic language, meter and rhyme. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1948–1958.

Lewis, D.; Zugarini, A.; and Alonso, E. 2021. Syllable neural language models for english poem generation. In *12th International Conference on Computational Creativity (ICCC'21)*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Loshchilov, I., and Hutter, F. 2017. Decoupled weight decay regularization. In *arXiv*.

Louis, A. 2020. BelGPT-2: a GPT-2 model pre-trained on French corpora. https://github.com/antoiloui/belgpt2.

Manurung, R.; Ritchie, G.; and Thompson, H. 2012. Using genetic algorithms to create meaningful poetic text. *Journal of Experimental & Theoretical Artificial Intelligence* 24(1):43–64.

Martin, L.; Muller, B.; Ortiz Suárez, P. J.; Dupont, Y.; Romary, L.; de la Clergerie, É.; Seddah, D.; and Sagot, B. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7203–7219. Online: Association for Computational Linguistics.

Poibeau, T.; Maignant, M.; Mélanie-Becquet, F.; Plancq, C.; Raffard, M.; and Roussel, M. 2020. Sonnet combinatorics with OuPoCo. In *Proceedings of the The 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, 133–137. Online: International Committee on Computational Linguistics.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1(8):9.

Shihadeh, J., and Ackerman, M. 2020. Emily: An emily dickinson machine. In *ICCC*, 243–246.

Toivanen, J.; Toivonen, H.; Valitutti, A.; and Gross, O. 2012. Corpus-Based Generation of Content and Form in Poetry. In *Proceedings of the Third International Conference on Computational Creativity*.

Van de Cruys, T. 2019. La génération automatique de poésie en français (automatic poetry generation in French). In *Actes de la Conférence sur le Traitement Automatique des Langues Naturelles (TALN) PFIA 2019. Volume I : Articles longs*, 113–126. Toulouse, France: ATALA.

Veale, T. 2016. The shape of tweets to come: Automating language play in social networks. *Multiple Perspectives on Language Play* 1:73–92.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, 38–45.

Yang, W.; Cheng, Y.; He, J.; Hu, W.; and Lin, X. 2016. Research on community competition and adaptive genetic algorithm for automatic generation of tang poetry. *Mathematical Problems in Engineering* 2016.

Yang, W.; Weng, W.; Chen, G.; and Jiang, Z. 2021. Elitist strategy of genetic algorithms for writing tang poetry. *International Arab Journal Of Information Technology* 18(4):604–610.

Zhang, H., and Zhang, Z. 2020. Automatic generation method of ancient poetry based on lstm. In *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 95–99. IEEE.

Zugarini, A.; Melacci, S.; and Maggini, M. 2019. Neural poetry: Learning to generate poems using syllables. In *International Conference on Artificial Neural Networks*, 313–325. Springer.

Adapting Transformer Language Models for Application in Computational Creativity: Generating German Theater Plays with Varied Topics

Lukas Wertz, Jonas Kuhn
University of Stuttgart
Institute for Natural Language Processing (IMS)
lukas.wertz@ims.uni-stuttgart.de

Abstract

Pre-trained transformer language models have been shown to generate human-like quality texts of different styles. In this study, we generate short drama dialogues in the style of German theater plays and adapt their content to various different topics using a simple fine-tuning scheme. We show that the generations keep the dramatic play structure while adapting large parts of their content to a target topic, effectively creating scenes from theater plays about a variety of subjects. We experiment with hyperparameters to find fitting fine-tuning configurations for various topic datasets as well as highlight how the generations adapt to the topics in a qualitative analysis. Our findings present a useful tool for computer assisted or fully autonomous creative writing. Furthermore, we motivate and explore the use of transformer language models in the context of computational creativity, highlighting the need for constrained and controlled language generation.

Introduction

This paper reports on a set of pilot experiments that we conducted in preparation of a possible integration of AI generated elements in an actual theater production. The output produced by recent transformer language models such as GPT-2 is often intriguingly natural. Yet when applying such models for language generation in a specific computational creativity context, there are typically additional constraints on the desired model output: in our pilot scenario, the generated text was for instance (i) supposed to follow the structural characteristics of dramatic text; in addition (ii), the text was supposed to revolve around a specific domain content. We argue that such constraints to the application of pre-trained language models are not a peculiarity arising from our application scenario, but reflect a general challenge that an integration of recent model types from Natural Language Processing (NLP) research into a computational creativity scenario faces.

Our preliminary experimental results on adapting transformer language models for creative language generation are thus not only informative for scenarios with a similar constellation of training and tuning resources; by reporting on experience from our pilot study we also hope to make a contribution to an open-ended (and presumably long-term) process of identifying suitable workflows and methodological set-ups for interdisciplinary work in the broader field of computational creativity research.

Motivation and Background

Many scenarios in which the acts of a human (or a group of humans) are commonly described as creative involve language production. Writing a novel, a poem, or a theater play is taken to involve creativity; but many other uses of language may as well. Take the example of giving a quick-witted response to an unpleasant interview question or to some remark that is considered inappropriate. Since language is ubiquitous in human activity – and it is comparatively easy to collect samples of language output and for instance process text corpora with the computer – it comes as no surprise that a lot of research on human or machine creativity targets creativity manifested in (some aspects of) text(s).

What is problematic however, in particular when the goal is to develop a systematic understanding of the processes underlying creativity, is the following: Language production (and hence text as its output) is a massively multi-layered phenomenon.

The multi-layered character of text. A highly diverse collection of knowledge spheres and contextual factors play together in the production of any element of text. Hence, pinpointing the role of a particular knowledge source in an account of creative behavior based on textual evidence is very hard. Since humans can effortlessly handle the network of cross relations among levels of language and text, a decision at one level will have consequences in multiple other levels in human-generated text. For instance, different ways of describing a certain action ("she warned him/she gave him a heads-up/she drew his attention to the fact that ...") may be truth-conditionally equivalent, but connotations, conventions in a particular text genre, domain-specific jargon, script knowledge about (culture-specific) scenarios etc. can make specific alternatives appear humorous, sarcastic, arrogant, mildly impolite, etc. Some of the most aesthetically appealing examples of creative language use keep most cross-level relations aligned with what is to be expected from conventions etc., but then break expectations (Takala 2005; Raby 2010) at a possibly subtle, but effective point. Creative language use thus plays with the reader's/audience's (mostly unconscious) knowledge about typical cross-dependencies of levels of language and text.

Consequences for computational creativity research. The rich interrelations between levels and connotations of language elements poses considerable challenges to systematic generative research. Controlled experiments manipulating certain text elements can easily be disrupted by side effects at entirely different text levels that for instance cause human readers to find passages unnatural.

For a long time, important subfields of computational creativity such as story generation (Gatt and Krahmer 2018; Gervas 2009), had therefore adopted the strategy of focusing on a particular text level for evaluation (and systematic description), e.g., the plot level. The surface realization of a formal plot description as an actual story does not aim to reach the aesthetic sophistication of human writing. Advances in the fields underline that this strategy of focusing on particular levels is effective for developing a better systematic understanding of particular elements of creative writing (without drawing into question that they interact freely in actual human creative performance) (Lehnert 1981; Holmes 1985; Papalampidi, Keller, and Lapata 2019).

Transformer language models. The developments in Natural Language Processing research on language modeling of the past 5-10 years call for a new assessment of the situation: transformer language models using hundreds of billions of parameters (Brown et al. 2020) and trained on gigantic collections of text apparently display a generative behavior that reflects many of the dependencies across text levels and relevant knowledge spheres. In typical application examples of completing a short text prompt, the models' choices in text production quite often seem to adhere to what for a human writer would be traced back to an intuition regarding connotations, genre convention and the other knowledge spheres listed above. Therefore, it is little wonder that the new generation of language models are finding many applications in a creative writing context (Bena and Kalita 2020; Ammanabrolu et al. 2019).

The solution? One might feel inclined to conclude that with transformer language models, the challenge from the multi-layered character of text for research on creativity has been overcome: The models are capable of generating stretches of text that are indistinguishable from human text. However, no matter whether one wants to employ such a model to enhance human creativity (in a co-creative scenario) or to use algorithmic models of creativity to advance our understanding of the processes underlying human creativity – the plain task of eloquently completing a given text prompt provides too little control. The language generator can "wander off" freely from the starting point and may take arbitrary "turns", most of which can be traced back to some explainable connection after the fact. But what is missing is even a slight element of goal-orientation. With all the difficulties in defining creativity, there is a far-reaching consensus that it not only involves an element of originality/novelty, but the product of creativity also needs to have a value of some kind (creativity as the production of "something original and worthwhile" (Sternberg, Sternberg, and Mio 2012)). This second element is not within the scope of the computational model of the process when the language model can wander off freely. For systematic research into creativity, this precludes the testing of specific hypotheses regarding creative processes (beyond the class of hypotheses that addresses only the role that experience and exposure to text collections that reflect certain conventions). In a pure creativity enhancement scenario, the inspiring effect of prompt-based generation alone may carry quite far, depending on the human maker's readiness to weed out fruitless output. But here too, exerting control over certain dimensions of the generated output could make the use of language models considerably more effective.

Desideratum. To make progress in the integration of transformer language models into computational creativity research and applications, we can hence identify a goal for the next years: a model architecture and methodological should be developed that is (i) based on current transformer language models with their ability to replicate the cross-level coherence of human language production, and (ii) at the same time allows for a constraining of several important dimensions of the generated text.

This paper reports on experimental work aiming to contribute to this goal. We start out with a pre-trained transformer language model and aim to constrain its generative behavior both in terms of text structure and in terms of the content domain that the text output is about. In a generalizable methodological set-up, it should be possible to characterize the two dimensions of constraining separately (i.e. the method should not only be applicable when there is a sufficiently large dataset for model tuning that happens to combine the two dimensions).

The computational work we report on in this paper grew out of pilot experiments conducted to have some tangible input for brainstorming sessions regarding the integration of AI generated elements in an actual theater production.

Related Work

From a computational point of view, automatic language generation has long been tackled as a task that would employ a pipeline architecture. First, knowledge structures such as dependency graphs or tables are used to form a plan of the events to describe (planning step). Then the the appropriate language is inserted via automatic grammars or slot-filling mechanisms (realization step). Such systems employ character goals (Meehan 1977), author goals (Dehn 1981) or underlying discourse states, (McKeown 1985) among other approaches (Callaway and Lester 2002) (Gervás et al. 2019). In recent years however, powerful language modeling approaches based on transformer deep neural networks (Vaswani et al. 2017) such as GPT-2 (Radford et al. 2019), GPT-3 (Brown et al. 2020) or T5 (Raffel et al. 2020) have shown to generate text of near human quality without the need of underlying knowledge structures (`https://ai.googleblog.com/2020/02/exploring-transfer-learning-with-t5.html`, `https://openai.com/blog/better-language-models/`). In order to achieve this level of knowledge, these language models are typically trained on millions of documents. However, while these models easily produce sophisticated text on their own, controlling the output can be difficult. One approach is to employ *Conditional Language Modeling* (Keskar et al. 2019) which prepends specific codes to text data. Using these codes during inference allows the language model to draw the generations from the part of the training data to which the code belongs. Other approaches apply the information at the self attention layer of the transformer network (Ziegler et al. 2019). The approach presented in this paper also shares similarities with (Dathathri et al. 2020)[1] who influence the gradient of the language model with keyword lists or a separate classifier in order to guide the language generation towards certain topics. Similarly (Pascual et al. 2021) use a simple method of guiding the language model towards semantically similar words of a desired topic. In many ways, the challenge in natural language generation today lies in reconnecting pre-trained language model with underlying knowledge structures (Chen et al. 2020a; 2020b; Peng et al. 2021).

Experiments

In our experiment, we build a text generation model using the GPT-2[2] transformer model from OpenAI (Radford et al. 2019). GPT-2 learns to predict the most probable next word in a sequence of words on a large collection of text. Pre-trained GPT-2 models have seen millions of documents and have been shown to produce several paragraphs of text almost indistinguishable from a human author (see **Related Work**). We leverage this learning power to first train a model that generates German theater plays. While these generations are already of fairly high quality, we inject these generated theater plays with new topics by further fine-tuning the GPT-2 generation models with additional data. We refer to this additional data as *topic corpus*. Our goal is to produce generations which keep the formal structure of a theater play (i.e. a sequence of speech acts that are exchanged between 2 or more characters) but change the topic of the play to that of the topic corpus. This way we attempt to constrain and guide the generation model towards a specific topic without changing the underlying structure of the learned language. We believe that by utilizing German language, our experiments have the additional merit of demonstrating the effectiveness of language modeling on languages other than English, which has already been extensively researched.

Datasets

The **Quadrama Corpus** (`https://quadrama.github.io/index.en`) is a machine readable collection of German theater plays from the late 18th and 19th centuries. The corpus contains many detailed annotations such as character relations, however we mostly make use of the plain text along with the annotations of the surface realizations of characters. We extract the text of each play token by token and mark the characters to which the text belongs by putting their names in capitalized letters followed by a colon (":"). We add a special token ($< |scene\_end| >$) at the end of every scene, which is later used in the generations. To form the plain text training corpus all scenes of all plays are concatenated into one text file. The final concatenated dataset contains 687 plays with a total of almost 14 million words. We refer to this dataset as *Quadrama*.

For fine-tuning the drama generation on a specific topic we use a variety of corpora, all in German language. We refer to each of these datasets as **topic corpus** throughout the experiments:

German Recipes Dataset. A collection of cooking recipes from a German recipe website available from *Kaggle*: `https://www.kaggle.com/sterby/german-recipes-dataset`. We concatenate all recipes to form the corpus, which we refer to as *recipe* corpus in the experiments. The *recipe* corpus contains 12190 recipes consisting of around 1.4 million words.

Horror Fanfiction. We create a small collection of stories categorized under *horror* from German website `https://fanfiction.de`. It should be noted that these stories do not contain popular media characters (as is common for fanfiction) but are entirely original. We concatenate all chapters of all stories into a single file to form the *horror-fanfiction* corpus. The corpus consists of 948 chapters with approximately 1 million words. **Expert Interviews**. This corpus contains a set of concatenated journalist interview transcriptions. The interview topics revolve around modern day issues concerning business, technology and role of artificial intelligence. We concatenate all interviews, including interviewer questions into a single file. In the experiments, we refer to this corpus as *expert-interview* corpus. This is our smallest corpus, containing 1242 utterances from 14 interviews and consisting of around 91000 words. description

Evaluation

Our goal in the evaluation is to get an idea of how well the generations adapt to the content of the desired topic while keeping the structure of theater plays. While we curate a number of generations for a qualitative analysis, we also devise a simple automatic evaluation scheme which uses a combination of three statistical properties of the generations with regards to the topic corpus. We preprocess each topic corpus by filtering stopwords and punctuation, lowercasing and stemming all words, creating a set of content words D we use to represent each topic corpus.

Given a collection of generations G, we first calculate the number of generated words that appear in D. For each $g \epsilon G$ we count how many of the generated tokens appear in D and average the count over all generations. We assume that the generations are thematically closer to the topic corpus when they use a higher number of content words. We refer to this

[1] We experimented with the approach in preliminary experiments but found the method to be difficult to tune and easily lead to repetitive generations.

[2] We choose to use GPT-2 over a newer model such as GPT-3 or T5 because of the relatively small size of GPT-2 (compared to its successors) and the high number of available pre-trained language models including a model trained on German texts.

measure as *content word frequency* (1).

$$content - word - frequency : \frac{\sum_{g\epsilon G} |\{w | w\epsilon G \wedge w\epsilon D\}|}{|G|} \quad (1)$$

$$topic{-}corpus{-}coverage : \frac{\sum_{g\epsilon G} \sum \frac{\{count(w)_C | w\epsilon g \wedge w\epsilon D\}}{|C|}}{|G|} \quad (2)$$

In addition to how many content words are used in each generation, we are also interested how frequent these words are in the topic corpus. For every $w\epsilon D$ we calculate the percentage of how often it appears in the set of tokens of the topic corpus *C*. We refer to this score as *corpus coverage*. For every *g*, we sum the corpus coverage of all content words that appear in *g* and then average over the whole generation set *G*, yielding a score we refer to as *topic corpus coverage*. We report *topic corpus coverage* as a percentage from 0 to 1. (2)

While the former two scores estimate the degree how much the generation model adapts to the topic corpus, we also want to make sure that we are not losing the text structure of theater plays. The nature of plays entails, that there are characters present in the text who speak the dialogue. We verify that this property holds in the generations by making use of the *Quadrama* annotations. In the *Quadrama* corpus, characters are written in capitalised letters followed by a colon (":"). Therefore, we can count how many speakers we find in a generation with simple surface matching. In the **Results** section we refer to this score as *number of speakers*.

Our quantitative evaluation approach gives us the possibility to investigate a large amount of generations automatically. In particular, we can verify to what extent the generation adapts its content words to the domain corpus. Overall, we omit an analysis of readability. Manual inspection of around 100 generated texts shows That quality of the generations is generally close to human level and the desired drama style is often difficult to read, even in the original drama corpus. We also decide not to evaluate coherence of the generated text, as that is not the focus of our experiment. We do however perform a qualitative analysis of two examples per domain corpus in the Section **Handpicked Examples**. We highlight both a successful topic adaption as well as a generation, where the play structure has been lost or the topic has not been integrated.

Setup

First, we fine-tune a pre-trained German GPT-2 model[3] on the *Quadrama* dataset (Section **Datasets**) for 3 epochs using *ADAM* optimizer with a starting learning rate of $5e^{-5}$. The resulting model is capable of generating drama text with consistent grammar in a very distinct language style (Figure 5). In order to incorporate domain specific content into the generated plays, we perform fine-tuning again using one of the topic corpora (see Section **Datasets**) for a single epoch with different learning rates. In particular, we investigate 3 learning rates: $5e^{-5}$, $5e^{-6}$ and $5e^{-7}$. We find that training with a learning rate higher than $5e^{-5}$ leads to overfitting and repetitive, stale generations.

It is common practice to provide a piece of text that the generation model then attempts to complete. This is also called **cue**. In the experiments we use very short pieces of text since we want the generations to be mostly dependent on the generation model. We also find that generating without a cue, the fine-tuned models will generally stick to the drama style language instead of incorporating the new information. As such, we provide a cue to the model that starts with a drama character and is then followed by one or two words from the topic corpus. We select words with a generally high frequency in the topic corpus (Section **Evaluation**) to serve as the generation model cue. For each learning rate, we fine-tune the *Quadrama*-model on the topic corpus and output 100 generations, using sampling decoding with a top *k* of 50. We generate until the $< |scene\_end| >$ token is reached up to a maximum of 100 tokens. For each topic corpus, we compare the output of the adapted generation models to the base *Quadrama*-model.

Results

Statistical Analysis

Figure 1 illustrates the results of the statistical evaluation across all generation experiments. Starting with the *recipe* topic corpus, we see that the fine-tuned generation model achieves significantly higher topic term frequency and corpus coverage when using the larger two learning rates ($5e^{-5}$, $5e^{-6}$). When using a learning rate of $5e^{-5}$, the model scores 30 words relevant to the topic in each generation on average, which is double the amount compared to using the *Quadrama only* model which was not fine-tuned on the *recipe* corpus. Similarly corpus coverage more than triples when using the larger two learning rates from 0.05 for the *Quadrama only* model to around 0.15 for the fine-tuned model. This signifies that the fine-tuned generations contain words which span around 15% of the *recipe* corpus. However, looking at the number of speakers we see that the improvements come at the cost of the play structure. Without fine-tuning on the *recipe* corpus, the model achieves 4 speakers per generation on average. This number decreases to 1 when using the larger two learning rates. We therefore assume that the play structure has been lost in most generations. We find that using the smallest considered learning rate $5e^{-7}$ yields the best compromise between play structure and topic integration. The fine-tuned model achieves on average 20 topic words which span around 7% of the topic corpus while keeping the average number of speakers around 3.

For the *horror-fanfiction* corpus, we find the overall best compromise between topic adaption and theater play structure when using the learning rate $5e^{-6}$. While the larger learning rate yields a higher number of topic words per generation it also decreases the number of speakers to an average of 1 per generation. The smallest learning rate preserves the number of speakers well at around 4 but

[3] using the language modeling code and *anonymous-german-nlp/german-gpt2* model freely available on huggingface: `https://huggingface.co`

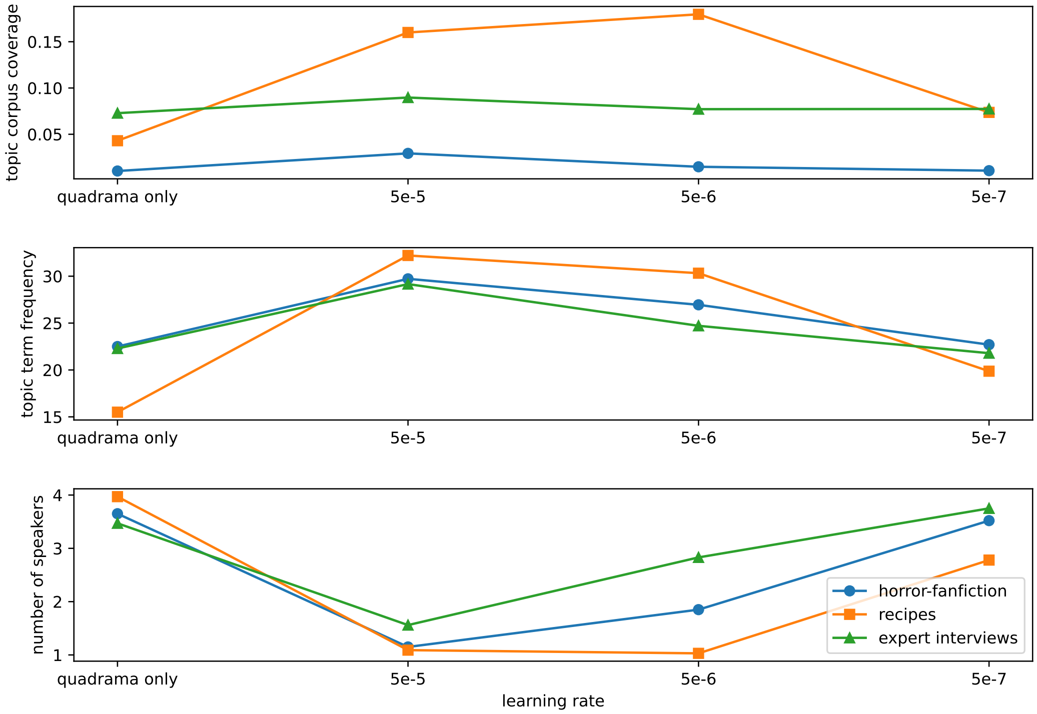


Figure 1: Statistical analysis of all generation models. Plots show, top to bottom *topic corpus coverage*, *topic term frequency* and *number of speakers* averaged over 100 generations from the generation model. The generation models considered are trained only on the *Quadrama* corpus (*quadrama only*) or received an additional fine-tuning step for 1 epoch with the listed learning rate on the x-Axis ($5e^{-5}$, $5e^{-6}$, $5e^{-7}$) on the respective topic corpus.

hardly affects the number of topic words (around 23, same as *Quadrama only* model) or their coverage the topic corpus (around 0.01, same as *Quadrama only* model).

Lastly, we generate theater plays with technical or business related topics by fine-tuning with the *expert-interviews* corpus. We find that the experiment behaves similarly to using the *horror-fanfiction* corpus. For the smallest learning rate $5e^{-7}$, the frequency of topic words per generation nor the coverage of the topic corpus improves upon using the *Quadrama only* model (around 23 and 0.8 respectively). The play structure is present, as can be seen by an average of 4 speakers per generation but we can assume that the topic is not integrated. Overall, we see that topic corpus coverage is not improved by any of the fine-tuned models, as it is already quite high for the *Quadrama only* model at around 0.08 compared to the other topic corpora. However, we find that when using the largest learning rate $5e^{-5}$ there are still slightly less than 2 speakers per generation on average while there is an improvement of 10 topic words per generation from around 15 to around 25 compared to the *Quadrama only* model. Therefore, we assume that the play structure is still present in many of the generations, if only with 2 speakers.

Handpicked Examples

We inspect all generated texts manually and select samples that illustrate the effect of topic integration in Figure 2, 3 and 4. We choose what we find to be the best fitting generations for the model trained only on the *Quadrama* corpus on the left. The highest quality generation for the model fine-tuned on the topic corpus is found on the right. The cue given to the model is marked in **bold**, phrases and words pertaining to the topic corpus (selected regarding to topic corpus words, see Section **Evaluation**) are marked in red and pointed brackets (<>).

Figure 2 shows that without fine-tuning on the topic corpus, the generation model accommodates the recipe domain to an extent. The characters aptly react to the presence of food items in the cue and mention **Zwiebeln**(*onions*) and **Pfeffer**(*pepper*), as well as **Salat**(*salad*) a number of times. The main language style of the generation is still that of a theater play, the food items seem to replace some of the subjects that are talked about. Overall, the presence of food items in the context seems to confuse the generation model which leads to the scene being generally difficult to understand and nonsensical in nature. Interestingly, the characters in this generation seem to actively move away from the topic of *salad*, as one character exclaims: **Wir haben uns zu sehr mit Salat beschäftigt.**(*We have concerned ourselves too much with salad*). The effect we see when fine-tuning on the *recipe-*

PRINZESSIN: Zuerst Tomaten<sauce, dann Zwiebeln>.
JAKOB: Nun? So?
SICHEL: Warum denn für < Salat>?
JAKOB: Dazu. Man kann jeden mit <Salat und Pfeffer>aufdecken. Und wenn man auch nichts Besseres bietet, als <Salat>, das macht mir sehr viel Spaß.
SICHEL: Wir haben uns zu sehr mit <Salat> beschäftigt.
JAKOB: Mir scheint, es freut mich.
SICHEL: Ist es denn möglich, daß sie dir...

PRINZESSIN: Zuerst Tomaten<sauce und Pfeffer> setzen, <dann die Gurken dazu und dann den Knoblauch dazu bis zum Rand fein hacken, bis sie den Salat putzen und damit die Gurken> schmieren!
FRAU VON WOLFF: So! So! Jetzt <erst die Gurken zusammen>pflegen und dann <den grünen Salat> in Topftöpfen festbinden <mit Pfeffer und Essig und den Wein eingießen und mit Salz und Pfeffer würzen>.
FRAU VON WOLFF: Aber lieber <das Salz ins Wasser>.

Figure 2: Generations for the *recipe* topic corpus. The left column shows generations with the model trained only on the *Quadrama*-corpus. The right column shows generations of models fine-tuned on the topic corpus.

PRINZESSIN: Im Schatten will ich hinscheiden.
Lulu.: Im <Schatten> will ich hinscheiden? Ich sehe des Vaters <Tod>!
EUSENLEITNER: Mit aller Kraft.
Lulu.: Und doch ist er mir entrissen – Und doch steht mein Mann –
TOLOMEO: Was soll das heißen? –
Lulu.: Er hat mich verlassen, aber ich darf nichts von ihm hören, wie das <Sterben eines Kindes>.
TOLOMEO: Wozu sollt ich hinauß?

PRINZESSIN: Im Schatten des Lichts, ist es <ein Dämon>?
DER KARAMUKAIN: Es ist <eine furchtbare Finsternis>. Ich wachte darauf, sah die Erde mit den Sinnen an. <Der Anblick dieses Untiers hat mich verwirrt>. Doch <im Schatten> des Lichts, ist es <ein Dämon>! Ich fühle mich nicht dort, fühle mich hier <gefangen>. Versteh mich nicht weiter von diesen Augen und auch nur von diesen verrannten Fühlen und von dem Glauben bei der Ankunft des Herrn.
(fine-tuned on *horror-fanfiction*, learning rate $5e-6$)

Figure 3: Generations for the *horror-fanfiction* topic. The left column shows generations with the model trained only on the *Quadrama*-corpus. The right column shows generations of models fine-tuned on the *horror-fanfiction* corpus.

corpus is much more pronounced: The characters dialogue essentially changes to recipes as the two characters in the fine-tuned example seem to actively talk about the preparation of a meal. The whole dialogue is structure in the style of a recipe with several ingredients being mentionend, such as **Knoblauch**(*garlic*), **Gurken**(*cucumber*) and **Essig**(*vinegar*). In addition, both characters also reference methods of preparation for these ingredients, such as **Salat putzen**(*clean the salad*) or **Wein eingießen**(*pour in Wine*). There is also still a degree of interaction between the speakers as the second character picks up the *cucumber* and salad mentioned by the first character and furthers the cooking instructions to now include seasoning. There are some incoherences: **Topftöpfen** would mean something like *potpots*, which does not have a clear meaning. Also **Tomatensauce und Pfeffer setzen** (*put tomato sauce and pepper*) is not a valid expression since the presence of the verb **setzen** would be highly confusing to a native German speaker in this context. In general though, incoherences seem particularly noticeable here as the recipe style dialogue contains explicit instructions that are easily understood and leave little room for interpretation compared to a more poetic style of language.

We illustrate two of the generations for the *horror* topic in Figure 3. Without additional fine-tuning on the *horror-fanfiction* corpus, the generation model already produces words that can be considered relevant to the topic, such as **Tod**(*Death*) or **sterben**(*to die*). However most of the language clearly sticks to the drama style. The word **hinscheiden**(*pass away*) for example is much more poetic and more typical of drama language than what we find in the topic corpus. The generation after fine-tuning on the *horror-fanfiction* corpus clearly adopts a more prosaic style, leading to a long text of the second character. We see, that the play structure is not as strong as it was before fine-tuning on the topic corpus. The language itself however reads much more like a scary novel using words such as **Finsternis**(*darkness*), **Dämon**(*demon*) or **Untier**(*monster*,*beast*). While the theater texts already contain a good amount of dramatic or even scary terms which lead to some adaption even without fine-tuning we find that the fine-tuned generations tend to fit the style of the topic corpus much better. Though this often leads to the characters talking in long monologues, effectively removing the theater play structure. As we already inferred from the statistical analysis, adapting to the *horror-fanfiction* corpus and keeping the drama language structure seems more difficult compared to for example the *recipe* corpus. A possible reason for this is that the chapters in the *horror-fanfiction* corpus tend to be a lot longer than a single recipe and as such, the generation model is incentivised to produce longer paragraphs of text. Another big difference is of course the author perspective which for the *recipe* corpus is fixed to an imperative, instructional style and in the *expert-interview* corpus is exclusively first person dialogue. Both of these language style naturally fit easily into theater plays, which consist mostly of discourse between several participants. The third person narration employed in many chapters of the *horror-fanfiction* corpus however is very rarely used in drama text. As such, we suspect the generation model has a harder time integrating much of the text in the corpus, causing many

PRINZESSIN: Der Zukunftsboom treibt noch immer den Geist des <Kapitals> herauf. Wenn es aber auf der weiten Welt dem Proletariat so lange nachhängt, so hat das keinen Grund : die Arbeiterschaft gewinnt die höchste Stellung.
BARBARA: Warum ist der <Weltverbesserer> so bieder und so tiefsinnig, mein Fürst?
ZERBINO: Sein Bruder ist vielleicht krank. Sein Vater hat ihn gesund und hat ihm alle Gesundheit verloren. Seit zwei Weltkriegen hat sein Vater über ihn zu sprechen, wie ...

PRINZESSIN: Der Zukunftsboom wird die <europäische Gesellschaft> bald reformieren können. Aber Sie haben hier eine Chance.
REDLICH: Ich habe hier <ein enormes Marktpotential> erkannt. Ich glaube auch, dass die <Industrie das Thema der Zukunft durchaus antizipieren> konnte, so dass man das nicht zu einer <Diskussion von Chancen oder Herausforderungen führt>. Und das ist auch die Grundlage dafür, dass wir es jetzt auch direkt mit den <ökonomischen, sozialen und politischen Dimensionen des Wandels> zu tun haben. Wir haben diese Chance nämlich in unserer <internationalen Debatte> ...
(fine-tuned on *expert-interviews*, learning rate $5e-5$)

Figure 4: Generations for the *expert-interview* topic. The left column shows generations with the model trained only on the *Quadrama*-corpus. The right column shows generations of models fine-tuned on the *expert-interview* corpus.

PRINZESSIN: Euer Hoheit, daß ich mich empfehle. Ein andres Exempel.
HERR VON BIEDERLING.: Ich bitte, auf wen ich mich beziehe :
DER JUNGE: Ich muß es zu ihm sagen, will dem Herrn von Biederling sagen, da ich dem Herrn auf dem Ball nicht allein zuhören soll.
FRAU VON MAUER.: Mein Herr, ich darf mich noch auf die Sache einigen.
DER DRITTE: Zum Glück kann ich mir meine Untersagung zur Unzeit überlegen, Ihr habt ja vor der Verlobung in der Galerie mit Herrn von Biederling gestanden.
HERR VON MAUER.: Ich weiß, daß Ihr Euch heute noch auf das Wort einläßt, weil –
DER JUNGE: Sie ist zu der Aufseherin gekommen, die in der Galerie wartet.

Figure 5: Generation from the German GPT-2 model fine-tuned on the *Quadrama* corpus. The cue given to the generation model is marked in **bold**.

generated texts to trail off into narrations rather than theater plays.

Figure 4 illustrates generation results using the *expert-interview* corpus. Again, we find that the model can adapt to the topic without seeing the topic corpus, albeit within the confines of its play context. The scene generated without fine-tuning on the topic corpus yields a conversation about politics, mentioning words like **Kapital**(*capital*),**Arbeiterschaft**(*working class*), **Proletariat** and **Weltkrieg**(*world war*), which are all topics that can reasonably occur in theater plays. Though these terms are technical and relate to finance and politics, they do not reflect the topics of the *expert-interview* corpus which deals more with modern day businesses and computer technology. After fine-tuning on the *expert-interview* corpus we find that the generation incorporates much more modern terms, such as **Marktpotential**(*market potential*) and **internationale Debatte**(*international debate*) which are not very typical of theater plays thus demonstrating a degree of topic integration that was not present before. It should be noted that this is the only experiment where we picked generations using the largest learning rate of $5e^{-5}$. While for the other two topics, this learning rate caused the play structure from the generations to be lost, here we can still find many generations with at least 2 speakers. This might well be because the *expert-interviews* corpus consists of dialogue-style language and as such, causes the model to retain this dialogue structure after fine-tuning.

Discussion

Selection of high quality Generations

First, we should note that there are many generations which do not exhibit the favourable properties of the ones shown in Section **Handpicked Examples**. Some generations do not include the topic at all, despite fine-tuning and the generation cue. Other generations that fit the desired topic stray to far from the structure of a theater play and as such do not introduce any speakers into the scene. In order to find high quality results manual inspection of the generations is necessary. We do find however, that the provided statistical analysis is helpful in selecting good generations. While we checked all generations when curating the best results we ended up finding the most promising generated scenes from the models that offered the best compromise between the number of speakers and the frequency of topic words. In addition, we believe that our approach works well in an assisted creative writing setting where the author has more control over text that is generated line by line. This way, the generation model can be used like a tool that inspires creative output, in our case theater plays with possibly unusual topics.

Quality of generated Scenes

We find that many of the generations lack coherence overall. Many of the spoken dialogues, while grammatically correct, are very hard or impossible to make sense of. We investigate a generated example fine-tuned only on the *Quadrama* corpus in Figure 5. While the general style of language is

PRINZESSIN: Euer Hoheit!
KÖNIG: Ja Prinzessin. Was gibt es?
PRINZESSIN: Nun, ich will die ganze Welt aufbieten, und Euer Hoheit wollen mich nicht in den Krieg stürzen.
KÖNIG: Ich bin ein alter Narr, und ich bin ein ehrlicher Mann ; ich habe mich mit den alten Menschen in Verbindung gesetzt.
PRINZESSIN: Das wäre ein Unglück, wenn Ihr mich nicht in den Krieg ziehen lassen würdet.

Figure 6: Generation from the German GPT-2 model fine-tuned on the *Quadrama* corpus. This generation was created in tandem with the generation model. Text pieces provided by the author are marked in bold, the remaining text is automatically generated.

very evocative of a classic German theater play, the actual content of the scene is harder to follow. We find some plot points in the scene though: Someone was waiting in a gallery (**Galerie**) before an engagement (**Verlobung**) and now an attendant (**Aufseherin**) is waiting there. It is not clear who is speaking to whom however, which makes constructing a narrative near impossible. We also find that the generation model greatly benefits from a longer context. Figure 6 shows a scene created by alternating between the human author and the generation model. We find that in Figure 6, existing characters are repeated more consistently. In addition, the princess (**PRINZESSIN**) character states a desire to go to a war in both of her generated passages, displaying a coherence that is not present in Figure 5.

Interestingly, we also find that after fine-tuning on a topic corpus, the generations generally show more coherence when they actually adapt to the topic and are easier to understand. This effect can also be observed in the generations presented in Section **Handpicked Examples**, for example in Figure 2, where the generation without fine-tuning on the topic corpus seems confused by the presence of particular words. We assume that the reason for this lies primarily in the fact, that the words which are relevant to the topic are very rare in the *Quadrama*-corpus and as such, the generation model is less certain on how to continue the generation. This can lead either to the generated words become more and more random or to the generation model starting to ignore the topic words. It should also be noted however, that the language present in the *Quadrama* corpus is generally very complex and often hard to understand even for native speakers. Theater plays employ a very distinct language style and often obscure details of characters motivations, actions and intentions within the dialogue. In addition, many plays in the corpus are more than on hundred years old and use a vocabulary that is very different to modern language. This is a possible reason why the GPT-2 generation model replicates the language style but struggles with generating a coherent narrative. Apart from providing longer contexts to the generation model, another possible way to possibly improve overall cohesiveness would be to use more data for more robust fine-tuning, avoiding possible overfitting. Another approach is to tackle the decoding process of the language model. There are decoding strategies that reportedly improve the coherence in generated content ((Holtzman et al. 2019)) and those methods will likely improve results in our experiments as well.

Conclusion

Across all experiments, we find that our fine-tuning approach can achieve integration of the desired topic without losing the structure of theater plays. In particular, we show that the generation models incorporate words and concepts that were not present before the fine-tuning on the respective topic corpus. Furthermore, we illustrate that these concepts are integrated into dialogue spoken by at least two characters, creating a mixture between the theater play structure and the respective topic. While there is still room for improvement, particularly in the coherence of generated texts and the fairly high selection effort, we conclude from our results that our approach generally achieves its goal of injecting a new topic into the existing language structure. Furthermore, our approach does not require abundant data or specialised annotations. Apart from the corpus of theater plays, topic corpora similar to the ones presented in Section **Datasets** can be easily acquired from openly available sources. In addition, we also show that such a topic corpus does not need to be particularly large. The smallest topic corpus we use is the *expert-interviews* corpus with less than one hundred thousand words and we still see a strong effect there. This is useful in practice, as training a transformer language model on too little data can quickly lead to overfitting and consequently causes uninteresting, often repetitive generations.

We propose to further experiment with different ways of encoding to improve the readability and coherence of the generations. We also encourage the use of our fine-tuning approach in creative writing settings, be it fully automatic or in co-operation with the generation model in order to try out unusual combinations of topics.

Author Contributions

Lukas Wertz - Idea, Revision of Full Paper, Related Work and References, Experiments, Discussion/Analysis
Jonas Kuhn - Introduction, Motivation and Background

Acknowledgements

This research was funded and supported by the Institute of Natural Language Processing (Institut für Maschinelle Sprachverarbeitung: **IMS**) at the University of Stuttgart.

References

Ammanabrolu, P.; Broniec, W.; Mueller, A.; Paul, J.; and Riedl, M. 2019. Toward automated quest generation in text-adventure games. In *Proceedings of the 4th Workshop on Computational Creativity in Language Generation*, 1–12. Tokyo, Japan: Association for Computational Linguistics.

Bena, B., and Kalita, J. 2020. Introducing aspects of creativity in automatic poetry generation.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language models are few-shot learners. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 1877–1901. Curran Associates, Inc.

Callaway, C. B., and Lester, J. C. 2002. Narrative prose generation. *Artificial Intelligence* 139(2):213–252.

Chen, W.; Chen, J.; Su, Y.; Chen, Z.; and Wang, W. Y. 2020a. Logical natural language generation from open-domain tables. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7929–7942. Online: Association for Computational Linguistics.

Chen, W.; Su, Y.; Yan, X.; and Wang, W. Y. 2020b. KGPT: Knowledge-grounded pre-training for data-to-text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 8635–8648. Online: Association for Computational Linguistics.

Dathathri, S.; Madotto, A.; Lan, J.; Hung, J.; Frank, E.; Molino, P.; Yosinski, J.; and Liu, R. 2020. Plug and play language models: A simple approach to controlled text generation.

Dehn, N. 1981. Story generation after tale-spin. In *IJCAI*, volume 81, 16–18. Citeseer.

Gatt, A., and Krahmer, E. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *J. Artif. Int. Res.* 61(1):65–170.

Gervas, P. 2009. Computational approaches to storytelling and creativity. *AI Magazine* 30(3):49.

Gervás, P.; Concepción, E.; León, C.; Méndez, G.; and Delatorre, P. 2019. The long path to narrative generation. *IBM Journal of Research and Development* 63(1):8:1–8:10.

Holmes, D. I. 1985. The analysis of literary style–a review. *Journal of the Royal Statistical Society. Series A (General)* 148(4):328.

Holtzman, A.; Buys, J.; Forbes, M.; and Choi, Y. 2019. The curious case of neural text degeneration. *CoRR* abs/1904.09751.

Keskar, N. S.; McCann, B.; Varshney, L. R.; Xiong, C.; and Socher, R. 2019. Ctrl: A conditional transformer language model for controllable generation.

Lehnert, W. G. 1981. Plot units and narrative summarization. *Cognitive Science* 5(4):293–331.

McKeown, K. R. 1985. Discourse strategies for generating natural-language text. *Artificial Intelligence* 27(1):1–41.

Meehan, J. R. 1977. Tale-spin, an interactive program that writes stories. In *Ijcai*, volume 77, 91–98.

Papalampidi, P.; Keller, F.; and Lapata, M. 2019. Movie plot analysis via turning point identification.

Pascual, D.; Egressy, B.; Meister, C.; Cotterell, R.; and Wattenhofer, R. 2021. A plug-and-play method for controlled text generation. *ArXiv* abs/2109.09707.

Peng, X.; Li, S.; Wiegreffe, S.; and Riedl, M. 2021. Inferring the reader: Guiding automated story generation with commonsense reasoning.

Raby, G. 2010. Improvisation and devising: The circle of expectation, the invisible hand, and RSVP. *Canadian Theatre Review* 143(1):94–97.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1(8):9.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* 21(140):1–67.

Sternberg, R. J.; Sternberg, K.; and Mio, J. 2012. *Cognitive psychology*. Cengage Learning Press.

Takala, T. 2005. Creativity as disruptive adaptation – a computational case study. In *HI'05 Sixth International Roundtable Conference Computational and Cognitive Models of Creative Design*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is all you need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Ziegler, Z. M.; Melas-Kyriazi, L.; Gehrmann, S.; and Rush, A. M. 2019. Encoder-agnostic adaptation for conditional language generation. *arXiv preprint arXiv:1908.06938*.

Extraction of Typical Story Plot Patterns from Genres within Japanese Popular Entertainment Works

Hajime Murai, Shuuhei Toyosawa, Takayuki Shiratori, Takumi Yoshida,
Shougo Nakamura, Yuuri Saito, Kazuki Ishikawa, Sakura Nemoto,
Junya Iwasaki, Shoki Ohta, Arisa Ohba, Takaki Fukumoto
Computer Science Department
Future University Hakodate
Hakodate, Hokkaido, 041-8655, Japan
h_murai@fun.ac.jp

Abstract

Several narratological studies have investigated plot patterns within specific story genres. However, these studies focused only on specific genres; thus, the common characteristics of general plot structures have not been recognized. In this study, cross-genre and subgenre comparisons of plot patterns were quantitatively performed based on common symbol sets to describe the plot structures. Common symbol sets for describing the plot structures were utilized for analyzing the plot structure to compare different genre plot patterns. The target genres and stories were selected based on sales rankings and popularity rankings for popular Japanese entertainment works. Typical plot patterns for each genre were extracted from the analyzed plot structures using the N-gram-based pattern extraction method. Inner structures of genres were extracted using hierarchical clustering. As a result, common plot characteristics and differences between genres, and interpretable subgenre plot patterns were extracted. Although the results of this paper are limited to the popular Japanese entertainment genre, the proposed method can apply to other stories and genres in different cultures.

Introduction

Automatic story generation based on computational literary studies has become a popular research method. There are various methods for automatic story generation, for instance, applying plot patterns based on traditional narratology (Gervas 2014), utilizing agent-based approaches (Binks, Roberts and Young 2016), or deep learning methods (Fan, Lewis and Dauphin 2019). It would be useful to clarify the characteristics and structures of existing stories for the development of these methods and to establish a method for human-like storytelling ability using artificial intelligence.

Narratology is the academic field dealing with the characteristics and structures of stories. There are several traditional studies on narratology. For instance, Propp insisted that 31 functional elements can compose about 300 Russian folktales about magic, based on analysis of story structure (Propp 1968). Furthermore, Campbell proposes that there is a common story structure within myths all over the world (Campbell 1949). Due to the influence of philosophical structuralism, the characteristics of those research targets were thought to be various structures (structures of plots, characters, and narratives); therefore, these research methods are called story structure analysis (Barthes 1968). Several studies have clarified that it is possible to extract the common plot structure of stories belonging to a specific genre by collecting and analyzing several such stories. Based on these old humanistic studies, recent research focusing on several specific genre stories has clarified that the quantitative and objective extraction of common plot structures can be executed using computational methods (Murai 2020; Suzuki, Sasaki and Hakamada 2018; Saito, Yoshida and Nakamura 2021; Iwasaki, Toyosawa and Ishikawa 2021; Oba, Ota and Amano 2021). In these recent studies, the plot structures were described as sequences of symbolized scenes or functions. The common plot structures of specific genres were extracted using quantitative methods for symbolized sequences.

However, these studies focused only on specific genres; thus, the common characteristics of general plot structures have not been recognized. Therefore, common symbol sets for describing the plot structures of several different genres have been developed (Murai, Toyosawa and Shiratori 2021a). Identifying common symbols across story genres enables a comparison of the characteristics of typical plot patterns of each genre. In addition, the internal structure of each genre has not yet been investigated. Moreover, the extracted typical patterns could become a foundation for automatic story-generation systems. In this study, cross-genre and sub-genre comparisons of plot patterns were quantitatively performed based on common symbol sets to describe the plot structures. Cross-genre comparison and sub-genre analysis would enable more sophisticated story generation for instance, genre combined story generation, and user's taste oriented detailed story generation.

Materials and Methods

Dataset and categories for plot analysis

Comics, games, and novels of several popular genres in modern Japanese entertainment culture were selected, based on the sales and popularity rankings, to compare different story genres (Murai, Toyosawa and Shiratori 2021a). The most popular selected genres were "Adventure," "Battle," "Love," "Detective," and "Horror." These are based on hypotheses that popular works should include some fascinating plot patterns for most people, and that typical plot patterns can be extracted quantitatively by gathering same genre works. To extract typical plot structures for each genre, works of combined genres (such as "love comedy") were eliminated, and popular short stories were selected based on the rankings. In cases where there were not enough popular short stories, popular long stories were divided into short stories based on the changes in the purpose of the stories' protagonists (Nakamura and Murai 2020).

Subsequently, the selected stories were divided into scenes, and scenes were categorized manually by researchers of narratology based on the functions of each scene as a story plot. The criteria of the story division are as follows (Murai, Matsumoto and Sato 2011):

- Physical transitions of places
- Elapsed time (except very short case for instance, a few seconds of silence)
- Appearance, exit, or movement to another place; birth and death of main characters
- End of explanation for readers (except very short case)

These criteria are based on traditional scene divisions in narratology.

After the scene division, the classification of each scene was performed based on the category table (Murai, Toyosawa and Shiratori 2021a; Murai, Toyosawa and Shiratori 2021b)

Table 1 depicts the number of analyzed stories and scenes in five genres. Table 2 lists the nine regions and 29 large categories of the plot elements. One story is depicted as a sequence of 29 scene types. For instance, if there is a story about a protagonist who encounters some man-made disaster, such as a battle, and finally defeats the ringleader behind it using a special given power (an example of a typical battle genre plot), that story can be depicted as a sequence of several large categories: "Disaster," "Ability improvement," "Confrontation," and "Everyday."

Moreover, 29 large categories were divided into 227 small categories. The relationships between the large and small categories are depicted in Table 3. In the categorization process, each scene was categorized based on small and large categories.

Each scene division and categorization process were performed by at least two individual analysts. When the results of the two analysts were different, they discussed which was better and decided on the final result. This type of collegial system (Someren, Barnard and Sandberg 1994) is often utilized in the literary analysis of the humanities field.

Table 1. Analyzed stories and scenes for each genre.

	Story	Scene	Average scenes per story
Adventure	226	1750	7.7
Battle	375	2994	8.0
Love	172	1604	9.3
Detective	134	1281	9.6
Horror	167	1484	8.9

Genre clustering based on plot sequence

To investigate the subgenre structure of each story genre, a clustering method was applied. The categorized scene sequences of the stories were clustered based on the Levenshtein distance of sequences composed of small categories. To avoid the effect of story length, the Levenshtein distance of two stories was divided by the shorter length of the two stories. More specifically, the hierarchical clustering from the Ward method was applied.

Plot pattern extraction

Each story genre was assumed to have a typical plot pattern. There are several methods to investigate frequently appearing serial symbol patterns, such as the N-gram or Markov chain. In this study, an N-gram-based pattern extraction method was applied (Saito, Yoshida and Nakamura 2021; Iwasaki, Toyosawa and Ishikawa 2021). In this algorithm, if the order of appearance of symbols is appropriate, non-continuous sequences are also calculated as N-gram pattern. The pattern extraction process was as follows:

1. N-gram distribution is computed based on plot sequences within target stories
2. Several patterns with high frequency are selected from N-gram
3. One element is added to the selected patterns under the condition that the added pattern appears as frequently as possible in the N-gram.
4. Several patterns in which the sum of the included N-gram's frequency is larger are selected.
5. Steps 3 and 4 are repeated until the pattern length becomes of the specified length which user can decide.

By applying this algorithm to a group of similar stories, a typical plot pattern can be extracted with an arbitrary length (Murai, Toyosawa and Shiratori 2021b).

A typical plot pattern of each story genre was extracted based on whole target genre stories for comparison. Moreover, to investigate the inner structure of each genre, typical plot patterns of clusters within genres were extracted based on stories within the target cluster.

An example of steps of 1 to 3 are also described in Figure 1. In Figure 1, 3-gram is calculated at first. In the next step, frequently appeared pattern(s) in 3-gram (in this example "ABC") was selected. In the third step, one symbol is added to the selected frequently appeared pattern "ABC" in order to include as match pattern as in calculated 3-gram. In this example, added pattern "ABCD" includes "ABC", "ACD" and "BCD."

Table 2. Large categories for cross-genre plot analysis.

Region	Large category	Description
Existence	Arrival	Encounter with the protagonist, including events such as birth and revival
	Leaving	Leaving from the story, including permanent departure such as death
	Change	Change in a character's attributes (e.g., swap, transform, and face change by plastic surgery)
Capability	Ability improvement	Positive change in a character's ability
	Ability decline	Negative change in a character's ability
Movement	Getting travel route	A character is able to travel
	Escape	Escaping from something (e.g., retreat, withdrawal, liberation, and prison break)
	Losing travel route	A character cannot move (e.g., losing transportation facilities, detention, kidnapping, and arrest)
Information	Search	Effort for obtaining information (e.g., exploration, survey, and research)
	Discovery	Disclosure of some information or hidden truth
	Misunderstanding	A character has a misunderstanding
	Doubt	A character notices something suspicious and has doubts
	Concealment	Some scenes about hiding information (e.g., concealment, disguise, and scam)
	External information	External information presentation for audiences through elements such as prologue and epilogue to explain about the world of the story
Regularity	Order, promise	It includes not only promise, transaction, and compliance, but also warning and prophecy.
	Violation	It includes crime, negligence, ignorance of warnings, and inattention.
	Intention, request	It includes scenes related to characters making decisions, that is, scenes involving wishing, request, persuasion, and invitation.
Intention	Completion of request	A scene that mainly consists of fulfilment of a request
	Failure of request	A scene that mainly consists of a failure or refusal to grant or fulfil a request
	Insanity	Situation wherein the character cannot control themselves (e.g., madness, confusion, and possession by evil spirits)
Relationship	Positive relationship	Positive changes in human relationships (e.g., conversion, reflection, reconciliation, expression of gratitude)
	Negative relationship	Negative changes in human relationships (e.g., quarrel, betrayal, arrogance, and disgust)
	Positive love relationship	Positive changes in human love (e.g., falling in love, confession of feelings, dating, and marriage)
	Negative love relationship	Negative changes in human relationships in the context of love (e.g., jealousy, broken heart, and divorce)
Influence	Aid	It includes many types of "help," such as rescue, nursing, assistance, encouragement, and sacrifice.
	Interference	It includes not only explicit interferences, but also acts that intentionally make the other person uncomfortable.
	Confrontation	Combat and competitions, including sports
Environment	Everyday	Scenes of ordinary everyday life
	Disaster	It includes not only natural disasters, but also accidents and mental crises such as severe depression

Table 3. Small categories for cross-genre plot analysis.

Large category	Small categories
Arrival	Arrival, Encounter, Resurrection, Birth, Making acquaintance, Reunited, Noticing a person
Leaving	Leaving, Exit, Death, Suicide, Exclusion, Sealed, Separation, Exorcism
Change	Change, Character's transformation, Replacement, Becoming another person, Memory loss, Pregnancy
Ability improvement	Ability improvement, Growth, Releasing sealed abilities, Becoming a companion, Getting back lost item, Gaining an item, Recovery, Recovery of physical condition, Healing, Improvement of social position, Enabling equipment, Cosmetic surgery
Ability decline	Ability decline, Withdrawal of a companion, Item loss, Stealing, Debility, Deterioration of physical condition, Illness, Injury, Social demotion, Incapacity, Sealed abilities, Memory loss
Getting travel route	Getting travel route, Opening a route, Acquisition of method for movement, Transportation, Moving
Escape	Escape, Retreat, Withdrawal, Extrication, Liberation, Disappearance
Losing travel route	Losing travel route, Detention, Kidnapping, Confinement, House arrest, Arrest, Blockade, Limitation of travel route, Change of travel route, Loss of method for movement
Search	Search, Exploration, Investigation, Expedition, Research, Experiment, Tracking, Vigilance
Discovery	Discovery, Disclosure, Confession, Exposure, Recovery of lost memory, Correct reasoning, Invention, Ingenuity
Misunderstanding	Misunderstandings, Misinterpretation, Mutual misunderstandings, Hallucinations
Doubt	Doubt, Mystery occurrence, Strange event, Disturbance, Misguided reasoning, Suspicion, Sign, Clue, Unaccounted
Concealment	Concealment, Deception, Takeover, Disguise, Fraud, Camouflage, Secret sharing, Ambush
External information	External information, Disclosure of world settings, Lessons learned, Recipes, Prologues, Epilogues, Afterglow, Side story
Order, promise	Order, Promise, Negotiation, Compliance, Warning, Notice, Prophecy
Violation	Violation, Stealing, Infidelity, Carelessness, Negligence, Ignoring warnings
Intention, request	Intention, Request, Determination, Declaration, Persuasion, Invitation, Acceptance, Seduction, Noticing a goal, Noticing a destination
Completion of request	Completion of request, Fulfillment of wish, Achievement of goal
Failure of request	Failure of request, Abandonment of wish, Failure to achieve the goal
Insanity	Insanity, Runaway, Possession, Confusion, Derangement, Stunned, Syncope, Drunkenness, Brainwashing, Enslavement
Positive relationship	Positive relationship, Conversion, Remorse, Reconciliation, Soothing, Acceptance of requests, Gratitude, Forgiveness, Hospitality
Negative relationship	Negative relationship, Quarrel, Betrayal, Arrogance, Disgust, Refusal of request, Provocation, Rebuke, Unkindness
Positive love relationship	Positive love relationship, One-sided love, Mutual love, Falling in love, Confession of love, Date, Dating, Marriage, Reconciliation of lovers, Physical relationship
Negative love relationship	Negative love relationship, Jealousy, Breaking up, Quarrel of lovers, Rejected love, Divorce, Prohibited romance
Aid	Aid, Protection, Rescue, Nursing, Encouragement, Sacrifice, Relief, Support
Interference	Interference, Enemy appearance, Intentional man-made disaster, Unreasonable demand, Intimidation, Annoying seduction, Bullying, Casting a spell, Revenge, Persecution
Confrontation	Confrontation, Battle, Competition
Everyday	Every day, Peace, Quiet, Daily event, Relaxation, Rest, Solution, Satisfaction, Praise
Disaster	Disaster, Damage, Natural disaster, Curse, Unintentional man-made disaster, Ordeal, Predicament, Disappointment, Despair, Shame, Regret, Dissatisfaction

Results

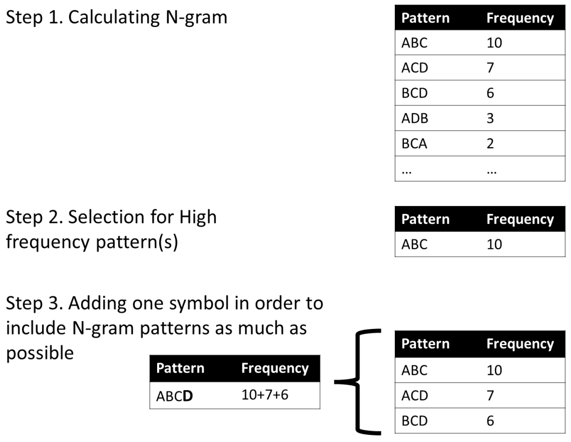
Hierarchical clustering of each story genre

The results of hierarchical clustering are presented in Figures 2–6. The dotted line indicates the cutting point of the dendrogram tree. Similarly, to compare each genre, five genres were divided into several clusters according to the number of stories included. Stories shorter than three elements were eliminated during the clustering process.

Typical plot pattern of each genre

To compare the typical plot patterns of each genre, an N-gram based plot pattern extraction algorithm [9, 10] was applied to stories of five genres. Table 4 shows the results of typical plot patterns based on a large category description of the plot sequences. Table 5 depicts the results based on the small category description. The number of scenes was set to eight, which was close to the average plot length.

Moreover, to extract typical plot patterns within genres and to investigate their internal structure, typical plot patterns of clusters within each genre were extracted based on small categories. Table 6 depicts the results of the adventure genre, Table 7 depicts the battle genre, Table 8 depicts the love genre, Table 9 depicts the detective genre, and Table 10 depicts the horror genre. The number of scenes in the typical plot patterns was set to 8 (this is near to average plot length).



Step 1. Calculating N-gram

Pattern	Frequency
ABC	10
ACD	7
BCD	6
ADB	3
BCA	2
…	…

Step 2. Selection for High frequency pattern(s)

Pattern	Frequency
ABC	10

Step 3. Adding one symbol in order to include N-gram patterns as much as possible

Pattern	Frequency
ABC**D**	10+7+6

Pattern	Frequency
ABC	10
ACD	7
BCD	6

Figure 1. An example of pattern extraction algorithm

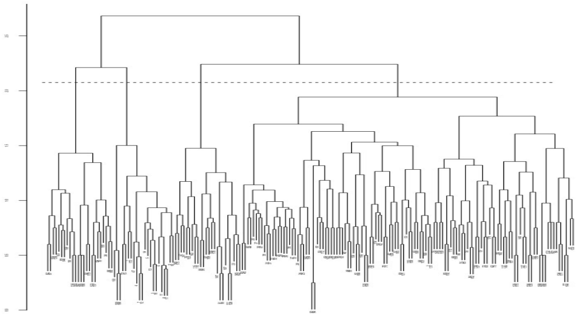


Figure 2. Clustering results of plot patterns in adventure genre

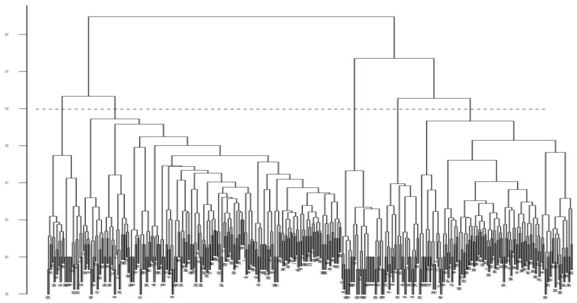


Figure 3. Clustering results of plot patterns in battle genre

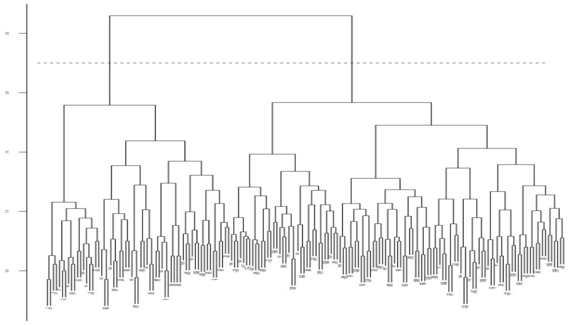


Figure 4. Clustering results of plot patterns in love genre

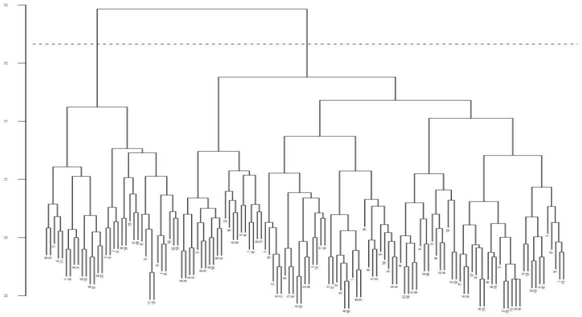


Figure 5. Clustering results of plot patterns in detective genre

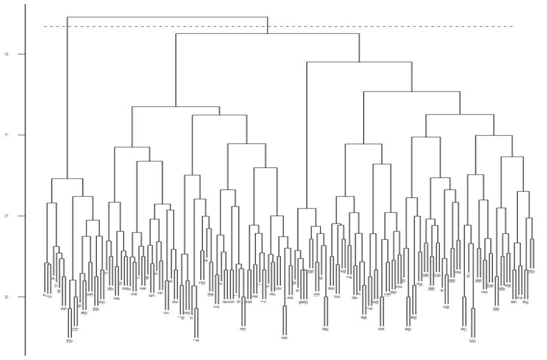


Figure 6. Clustering results of plot patterns in horror genre

Table 4. Genre-typical plot patterns based on large categories for cross-genre plot analysis.

	Adventure	Battle	Love	Detective	Horror
1	Search	Arrival	Positive love relationship	Discovery	Arrival
2	Discovery	Interference	Negative love relationship	Intention	Intention
3	Losing travel route	Aid	Arrival	Search	Discovery
4	Intention	Confrontation	Aid	Arrival	Exit
5	Ability improvement	Exit	Positive love relationship	Discovery	Completion of request
6	Getting travel route	Discovery	Discovery	Interference	Arrival
7	Search	Interference	Negative love relationship	Search	Intention
8	Discovery	Aid	Positive love relationship	Discovery	Discovery

Table 5. Genre-typical plot patterns based on small categories for cross-genre plot analysis.

	Adventure	Battle	Love	Detective	Horror
1	Blockade	Battle	Making acquaintance	Discovery	Request
2	Opening a route	Exclusion	Date	Investigation	Leaving
3	Expedition	Enemy appearance	Falling in love	Encounter	Completion of request
4	Arrogance	Interference	Confession of love	Exposure	Gaining an item
5	Battle	Incapacity	Positive love relationship	Discovery	Making acquaintance
6	Battle	Support	Dating	Investigation	Discovery
7	Death	Battle	Disclosure	Correct reasoning	Request
8	Opening a route	Exclusion	Date	Confession	Completion of request

Table 6. Genre-typical plot patterns of clusters in adventure genre based on small categories.

	Cluster 1	Cluster 2	Cluster 3	Cluster 4
1	Expedition	Blockade	Reunited	Gaining an item
2	Blockade	Arrogance	Exposure	Reunited
3	Provocation	Battle	Search	Change
4	Battle	Battle	Nursing	Blockade
5	Gaining an item	Death	Exposure	Battle
6	Opening a route	Opening a route	Confusion	Support
7	Expedition	Rescue	Request	Change
8	Battle	Disclosure	Rebuke	Battle

Table 7. Genre-typical plot patterns of clusters in battle genre based on small categories.

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
1	Interference	Disturbing	Enemy appearance	Battle	Reunited
2	Support	Completion of request	Quarrel	Enemy appearance	Battle
3	Battle	Reunited	Detention	Interference	Battle
4	Exclusion	Exposure	Intimidation	Incapacity	Reunited
5	Exposure	Request	Support	Support	Betrayal
6	Interference	Exposure	Aid	Gaining an item	Battle
7	Support	Nursing	Battle	Battle	Incapacity
8	Battle	Ordeal	Relief	Exclusion	Negative relationship

Discussion

Table 4 indicates that each genre pattern includes characteristic plot elements. For instance, Adventure and Detective genres include “Search” and “Discovery” twice. However, the difference of “Search” between adventure and detective genre is unclear. Conversely, Table 5 indicates that “Expedition” appears in the Adventure genre and “investigation” appears in the Detective genre. Therefore, a small category would be appropriate for extracting the differences between typical plot patterns in each genre. However, to extract the commonality of different genres, large category-based plot patterns would be appropriate.

Considering the similarities between different genres, each one has “Discovery” based on a large category. Therefore, the common plot structure for the five genres would be to disclose new information to the reader with surprise. Moreover, “Arrival” of new characters is also a common function for several genres. However, the identity of the new character may differ depending on the genre.

From the viewpoint of differences, there are various differences between genres based on small categories, even if those plot elements are included in the same large category.

These differences can be investigated by analyzing the internal structures of genres. Typical plot patterns based on clustered stories in each genre indicate that there are various subtypes within a genre. Table 11 shows a manual interpretation of the extracted plot patterns for each cluster. These plot patterns can be interpreted as typical patterns within a specific genre.

Conclusions

Based on the category table for cross-genre plot analysis and five genres of plot data sets of Japanese popular entertainment stories, cross-genre comparisons for typical plot patterns were performed in this study. As a result, common plot elements were extracted and differences between genres were depicted. Moreover, genre stories were clustered using the hierarchical clustering method, and typical plot patterns of each cluster were also extracted. The extracted plot patterns of clusters can be interpreted as typical story types within the target genres. Since the resulting patterns can be thought to be eligible detailed characteristics of the target genres, those features would be applicable to develop more sophisticated story generation algorithms. In addition to the plot patterns, if the patterns within the relationships and the roles of story characters can be quantitatively extracted, it would become the basis for more impressive story generation ability in the future.

The results of this study are confined to the popular Japanese entertainment genre, and the number of analyzed stories was about only 1,000. Therefore, the obtained clusters and patterns cannot be claimed to be general or universal. However, the method proposed in this paper can be applied to other stories and other genres in different cultures by expanding the category table appropriately. Moreover, it could be possible to investigate more detailed genre inner structures by adding many more stories to the plot data set.

Acknowledgements

This research was funded by NEDO “Development of interactive story-type content creation support platform” project.

Author Contributions

HM substantially contributed to the study conceptualization and the manuscript drafting. ST, TS, TY, SN, YS, KI, SN, JI, SO, AO and TF significantly contributed to data analysis and interpretation. All authors approved the final version for submission.

References

Barthes, R. 1968. *Elements of Semiology.* Hill and Wang, New York, USA.

Binks, A. A.; Roberts, D. L.; and Young, R. M. 2016. Summarizing and Comparing Story Plans. In *Proceedings of Workshop on Computational Models of Narrative 2016*, 107–123.

Campbell, J. 1949. *The Hero with a Thousand Faces*. Pantheon Books, USA.

Fan, A.; Lewis, M.; and Dauphin, Y. 2019. Strategies for Structuring Story Generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics 2019*, 2650–2660.

Gervas, P. 2014. Reviewing Propp’s Story Generation Procedure in the Light of Computational Creativity. In *Proceedings of The Society for the Study of Artificial Intelligence and Simulation for Behaviour*, Goldsmiths, University of London.

Iwasaki, J.; Toyosawa, S.; Ishikawa, K.; Ohta, S.; and Murai, H. 2021. Cross-genre Plot Analysis of Detective and Horror Genres. In *Proceedings of JADH Annual Conference,* 106–110.

Murai, H.; Matsumoto, N.; Sato, C.; and Tokosumi, A. 2011. Towards the numerical analysis of narrative structure: The characteristics of narrative structure within the short stories of Shinichi Hoshi. *JSIK*, 21 (1):6-17 (in Japanese).

Murai, H. 2020. Factors of the Detective Story and the Extraction of Plot Patterns Based on Japanese Detective Comics. J. *Jpn. Assoc. Digit. Humanit.* 2020, 5(1):4–21.

Murai, H.; Toyosawa, S.; Shiratori, T.; et al. 2021. Dataset Construction for Cross-genre Plot Structure Extraction. In *Proceedings of JADH Annual Conference*, 93–96.

Murai, H.; Toyosawa, S.; Shiratori, T.; et al. 2021. Extraction of factors that characterize the structures of plot within each story genre. In *Proceedings of IPSJ SIG Computers and the Humanities Symposium 2021*, 16-23 (in Japanese).

Nakamura, S. and Murai, H. 2020. Proposal of a method for analyzing story structure of role-playing games focusing on quests structure. In *Proceedings of IPSJ SIG Computers and the Humanities Symposium*, 149–156 (in Japanese).

Oba, A.; Ota, S.; Amano, T.; et al. 2021. Construction of a game system that automatically generates integrated quest scenarios, sound effects, and characters which focus on immersiveness. In *Proceedings of the 35th Annual Conference of the Japanese Society for Artificial Intelligence 2021*, 3D1-OS-12a-02 (in Japanese).

Propp, V. 1968. *Morphology of the Folk Tale*. University of Texas Press, USA.

Saito, T.; Yoshida, T.; Nakamura, S.; et al. 2021. Basic Plot Structure in the Adventure and Battle Genres. In *Proceedings of JADH Annual Conference*, 97–100.

Someren, M. W.; Barnard, Y.; and Sandberg, J. 1994. *The Think Aloud Method*, London, UK: Academic Press.

Suzuki, R.; Sasaki, S.; Hakamada, S.; et al. 2018. Examination and development of an integrated system for automatic generation of plots and visualization of scenes. *SIG Technical Reports 2018*, 2018-EC-50 (28):1–8 (in Japanese).

Table 8. Genre typical plot patterns of clusters in love genre based on small categories.

	Cluster 1	Cluster 2
1	Date	Prologues
2	Positive love relationship	Making acquaintance
3	Mutual misunderstandings	Request
4	Falling in love	Rescue
5	Confession of love	Positive love relationship
6	Dating	Making acquaintance
7	Jealousy	Date
8	Date	Disclosure

Table 9. Genre-typical plot patterns of clusters in detective genre based on small categories.

	Cluster 1	Cluster 2
1	Discovery	Request
2	Investigation	Exposure
3	Encounter	Investigation
4	Exposure	Correct reasoning
5	Discovery	Confession
6	Investigation	Discovery
7	Correct reasoning	Escape
8	Confession	Battle

Table 10. Genre-typical plot patterns of clusters in horror genre based on small categories.

	Cluster 1	Cluster 2
1	Request	Hospitality
2	Leaving	Gaining an item
3	Completion of request	Quiet
4	Gaining an item	Discovery
5	Making acquaintance	Arrival
6	Discovery	Hospitality
7	Request	Leaving
8	Completion of request	Discovery

Table 11. Interpretation of each story type of extracted plot patterns based on cluster.

Genre	Cluster	Story type
Adventure	1	Expedition for extermination of an enemy
	2	Expedition for finding a truth
	3	Resolution of some illness or injury
	4	Problem and solution about a change in a characters' attribute
Battle	1	Resistance against repeated interference
	2	Guidance to an ordeal
	3	Relief for victims
	4	Victory by obtaining special power
	5	Detection and punishment against traitors
Love	1	Resolving misunderstandings between lovers
	2	Prince Charming
Detective	1	Encounter to a murder case
	2	Request for resolving a murder case
Horror	1	Request from a ghost
	2	Requital of a favor by a ghost

Training GPT-2 to represent two Romantic-era authors: challenges, evaluations and pitfalls

Piotr Sawicki[1], Marek Grześ[1], Anna Jordanous[1], Dan Brown[2], Max Peeperkorn[1]
[1] School of Computing, University of Kent, Canterbury, UK
[2] Cheriton School of Computer Science, University of Waterloo, Canada
P.Sawicki@kent.ac.uk, M.Grzes@kent.ac.uk, A.K.Jordanous@kent.ac.uk, Dan.Brown@uwaterloo.ca, mp770@kent.ac.uk

Abstract

Poetry generation within style constraints has many creative challenges, despite the recent advances in Transformer models for text generation. We study 1) how overfitting of various versions of GPT-2 models affects the quality of the generated text, and 2) which model is better at generating text in a specific style. For that purpose, we propose a novel setup for text evaluation with neural networks. Our GPT-2 models are trained on datasets of collected works of the two Romantic-era poets: Byron and Shelley. With some models, overfitting manifests by producing malformed samples, with others, the samples are always well-formed, but contain increasingly higher levels of n-grams duplicated from the original corpus. This behaviour can lead to incorrect evaluations of generated text because the plagiarised output can deceive neural network classifiers and even human judges. To determine which model is better at preserving style before it becomes overfitted, we conduct two series of experiments with BERT-based classifiers. Overall, our results provide a novel way of selecting the right models for fine-tuning on a specific dataset, while highlighting the pitfalls that come with overfitting, like reordering and replicating text, towards more credible creative text generation.

Introduction

Contemporary text-generation systems can create output whose surface features strongly resemble the source materials upon which they are trained. Such generative systems can have credibility in computational creativity research as they can be demonstrated to possess knowledge and produce novel and valuable results in a directed fashion (Ventura 2016). As described below, there is a growing body of work that has recently been emerging in this direction in Natural Language Processing (NLP) research, via the growing popularity of OpenAI's GPT (Radford et al. 2018; 2019; Brown et al. 2020). While GPT-based systems for stylistic reproduction have attracted some criticism (Falk 2021; Floridi and Chiriatti 2020), in general, their results have been impressive and deserve further attention in computational creativity research (Dale 2021; Köbis and Mossink 2021).

Soon, these systems could generate new works in the style of authors from previous eras, perhaps even inspired by current events or social movements. However, before this can become reliably possible without relying on human supervision to cherry-pick the results, we need to learn how well these new transformer-based systems can be trained, and what pitfalls exist with them. In particular, we need to know what are the differences in performance between various versions of the models, to allow for optimal selection. Here, we describe some steps toward these aims.

Ideally, we would like to conduct a large scale human evaluation of GPT-2 produced text, but such evaluations are prohibitively costly and difficult to organize for most researchers, therefore in this study we are focused almost entirely on automated evaluations. For the first objective of this study—detection of over-training of GPT-2 models—we perform a visual evaluation of the samples to watch for malformed text, and then we perform the BLEU (Papineni et al. 2002; Yu et al. 2017) evaluation of the samples to watch for excessively high levels of similarity (on the n-gram level) of the samples to the original dataset. For the second objective, which is to investigate which GPT-2 model performs best at the task of generating text in specific authors' style, we use BERT (Devlin et al. 2018), which is currently state-of-the-art in text classification, to identify texts that appear closer to the source material than to the output of GPT-2 models.

Poetry generation has long been an area of interest in computational creativity research (Lamb, Brown, and Clarke 2017; Oliveira 2009; 2017). Previous work includes the use of machine learning (Das and Gambäck 2014; Loller-Andersen and Gambäck 2018; Rahman and Manurung 2011), mining of corpora or other source material as inspiring examples for stylistic reproduction (Gervás 2011; Lamb and Brown 2019; Rahman and Manurung 2011; Toivanen et al. 2014) as well as approaches such as expert-based distributed systems (Corneli et al. 2015; Misztal and Indurkhya 2014), the use of constraints (Rashel and Manurung 2014; Toivanen et al. 2013), evolutionary approaches (Manurung 2004) and knowledge-based/linguistic models (Hämäläinen 2018; Oliveira and Cardoso 2015; Veale 2013). Style imitation systems have also attracted attention in text generation (Alvarez Cos, Perez y Perez, and Aliseda 2007) and other domains (Ens and Pasquier 2018; Pachet and Roy 2014), though we note that attention has also been paid to the creativity required to deviate from a given style (Elgammal et al. 2017).

The growing attention being paid to GPT-based approaches in NLP research is beginning to get replicated in computational creativity. Here, we could mention a few notable examples where GPT-2 or BERT were applied to poetry generation: (Liao et al. 2019) have fine-tuned GPT-2 to generate Chinese classical poetry, (Köbis and Mossink 2021) have conducted an extensive human evaluation of GPT-2 generated English poetry, (Li et al. 2020) have experimented with applying rigid constraints in generation of both Chinese and English poetry, (Wöckener et al. 2021) have analysed the problems with maintaining rigid stylistic constraints in poetry generation while using RNN and GPT-2, and (Nikolov et al. 2020) and (Oliveira 2021) have explored a transformative, BERT-based approach to lyrics generation. Lyrics were also generated from GPT-2 and evaluated using BERT in (Wesek 2019).

The scope of this study is focused on poetry generation in a general language style of a specific author, learning from a corpus of poetry by two highly-regarded English poets: Lord Byron (1788-1824) and his contemporary Percy Bysshe Shelley (1792-1822). Here, we tackle poetry generation by fine-tuning GPT-2 models on the collected works of both authors.

GPT-2 and BERT models

The GPT and BERT models are derived from the Transformer architecture (Radford et al. 2018; Devlin et al. 2018), which is a form of an Encoder-Decoder model, where RNN or LSTM networks have been replaced by multiple layers of attention mechanisms (Bahdanau, Cho, and Bengio 2014; Vaswani et al. 2017), thus allowing all input to be processed simultaneously by dispensing with sequentiality. The original Transformer model followed the Encoder-Decoder architecture because it was intended for machine translation, where we first have to encode the source language sequence, and then decode it into the target language sequence (Sutskever, Vinyals, and Le 2014). Most other NLP tasks, however, do not require this kind of setup, and subsequently, the Encoder and Decoder blocks started to be used separately. The Decoder block was first developed by OpenAI into the Generative Pre-trained Transformer (GPT), and soon later the Encoder block was developed by Google into the Bidirectional Encoder Representations from Transformers (BERT). The first editions of GPT were released in two versions: small and medium, and have managed to advance the benchmarks on many NLP tasks (Radford et al. 2018). BERT was also released in two versions, which roughly matched the size of the small and medium GPT models to facilitate comparison. BERT has proven superior to the matching GPT models in Natural Language Understanding, while GPT excelled in Natural Language Generation (Devlin et al. 2018). This was expected because of the specific differences between the architectures of the Encoder and Decoder transformer blocks.

While the original Transformer (i.e. the translation machine) required separate training for each language pair, both GPT and BERT follow the transfer learning paradigm (Radford et al. 2018; Devlin et al. 2018). Both of them are first pre-trained on the large corpora of text (ranging from 5GB to 800GB, depending on the version). This initial training is very demanding in terms of time and the required hardware. The model can then be used "out of the box", or can be fine-tuned for a specific task and that is where transfer learning actually comes to play. The fine-tuning process is much faster and requires much less powerful hardware. After fine-tuning, the model can be used for a destined downstream task, using additional layers, which are referred to as "heads", that accomplish those tasks (Radford et al. 2018; Devlin et al. 2018).

The consecutive GPT versions use the same general architecture, but with much larger numbers of layers and "attention heads" (attention mechanism working in parallel). They are also trained on increasingly large datasets. Currently, the latest version of OpenAI's GPT is GPT-3; however, it can only be used through OpenAI's API. While the use of the base GPT-3 models through an API is free, fine-tuning of the model (at the time of writing this paper) has to be paid for. There exist other large language models, already exceeding the size of GPT-3, for example: Gopher (Rae et al. 2021), Megatron-Turing NLG (Smith et al. 2022), and Jurrasic-1 (Lieber et al. 2021), which, if available to the public at all, can only be used via their APIs. The hardware requirements of these models are way above of what researchers at most universities can access. Here, we should also mention the models released by EleutherAI: GPT-J-6B (Wang and Komatsuzaki 2021) and GPT-NeoX-20B (Black et al. 2022). However, their hardware requirements, while smaller than those of the models mentioned above, still exceed the hardware we can access.

This study was therefore carried out using GPT-2, which we could fine tune on our hardware. Applying GPT-3 and other large-scale models to poetry generation is left for future work. It is not unreasonable to expect that the results we obtained using GPT-2 will translate to larger language models when they become more accessible.

At present, there are two different applications of GPT-2 available. The original edition of GPT-2 is by OpenAI (Radford et al. 2019; OpenAI 2021). The source code of the interface for this version was later propagated with some changes by (Shepperd 2021), and as such it was used in research, for example (Lee 2019; Lee and Hsiang 2020b; 2020a). The second application is in the Transformers library (Transformers Documentation 2019; Wolf et al. 2019). This edition introduced significant changes to the code of the interface, making the process of fine-tuning and generation easier. We are using both applications in our experiments.

The GPT-2 models from the Transformers library that are used for text generation are referred to in their library as "Language Modelling Head" models (Transformers Documentation 2019), therefore in order to distinguish them from OpenAI models, in this paper, we refer to them as "LMH" models, while the OpenAI versions are referred to as "Regular". These two implementations differ significantly. The LMH models have a standard learning structure of training epochs, where each epoch has a specific number of steps depending on the size of the dataset. The Regular models do not use epochs in training. Additionally, training of the

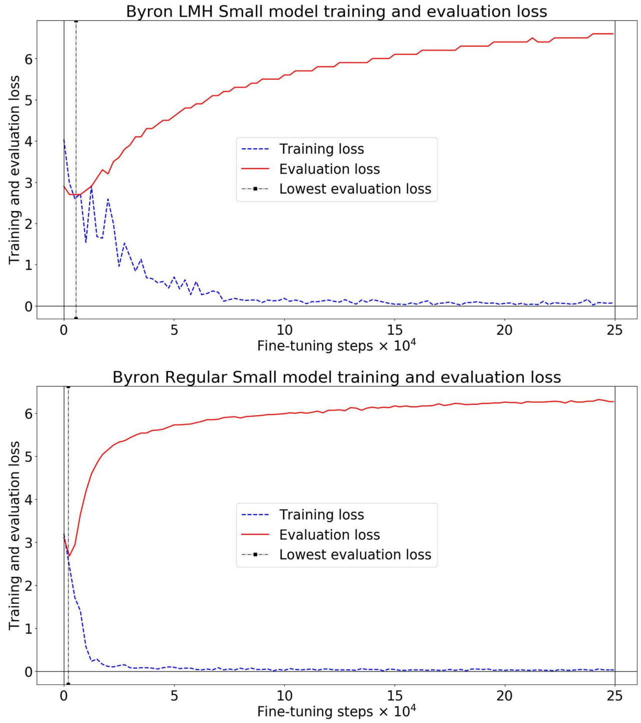


Figure 1: Training and evaluation loss for the LMH Small model (Top) and the Regular Small model (Bottom) fine-tuned for 250K steps on the dataset of Collected Works of Lord Byron.

LMH models is much faster per training step as compared to the Regular models. The Regular models are released in four versions: Small (124M parameters), Medium (345M parameters), Large (774M parameters) and XLarge (1558M parameters). The LMH models, at the time of writing this paper, were released only with Small, Medium and Large versions, with the same number of parameters as the respective Regular models. Due to hardware limitations, we only fine-tune the Small and Medium models, and as a result, we use four models in total for each task: LMH Small, LMH Medium, Regular Small and Regular Medium. Experiments with the Large and XLarge models are left for future research. It has to be noted that, at this point, we are not experimenting with adjusting hyperparameters, neither during fine-tuning nor during sample generation. The default *top_k* for both models is 50, default *temperature* is 1, default *top_p* is 1 for Regular models, and 0.9 for LMH models. For consistency, we have set *top_p* value for LMH models to 1. The train/test split of the datasets is 70/30.

Figure 1 shows the training loss and evaluation loss for the Regular Small and LMH Small models fine-tuned on the Byron dataset for 250K steps (results for medium models and for models fine-tuned on the Shelley dataset are very similar, and therefore are not presented here). We can see that the lowest evaluation loss is achieved very early in the fine-tuning process: for LMH models, this occurs around 5000 fine-tuning steps, and for the Regular models even sooner, around 1700 steps. We believe this is because the datasets used for fine-tuning are relatively small, and the models become overfitted fairly early. In this study, we investigate whether the point of the lowest evaluation loss is optimal for early stopping of the fine-tuning process, and to get deeper insights into the behaviour of the GPT-2 models, we evaluate the actual quality of the generated samples. To that end, we conduct a number of evaluations of samples generated at specific checkpoints. This will be further described in the later sections.

Author	Org size	Length	Final size	Length
Byron	7.2 MB	183643	2.4 MB	62947
Shelley	2.3 MB	59207	0.98 MB	29151

Table 1: Details of the datasets including original size, original length (lines), pre-processed size, pre-processed length (lines).

Data preparation

Original datasets

Our main interest is generation and evaluation of poetry in the style of a specific author. For that purpose we have chosen two Romantic-era poets: Lord Byron and Percy Bysshe Shelley. The datasets for both of them were created from their collected works, downloaded from Gutenberg.org (Project Gutenberg 2020), by removing all introductions, forewords, footnotes, generic Gutenberg text at the end of the file, replacing all extended Latin characters with a closest matching ASCII character (for example" Ã" is replaced with "A", "á" with "a", etc.). Sequences of multiple blank lines in the original text were replaced by a single blank line. We have removed all metadata, i.e., page numbers and the end of line verse numbers. We have left the poems' titles and chapter numbers, since they contribute to author's "style", but also to preserve the separation of individual poems. Being aware of the destructive impact that artefacts of poor pre-processing of data can have on the output of GPT-2, we have paid particular attention to the task of data preparation. Both the original and the pre-processed datasets can be found on our online repository[1].

Additionally, we have removed all plays, thus leaving only poetic works. The purpose of this pre-processing was to leave only the poetic text written by the authors themselves.

Setup 1 for visual and BLEU evaluations

We fine-tune all four GPT-2 models used in this study (Regular Small, LMH Small, Regular Medium, LMH Medium) on both datasets (Byron and Shelley). For the visual and BLEU evaluation in Experiments 1 and 2, we fine-tune the GPT-2 models for 250K steps, generating 100 samples at each 10K steps interval. The samples are generated with a length of 1000 tokens (the maximum sample length for these models is 1024 tokens). We generate only 100 samples at

[1] Our datasets and example outputs generated by GPT-2 are available at:
`https://github.com/PeterS111/GPT-2-for-Byron-and-Shelley`

each checkpoint because of the time it takes to generate full-length samples (for example the LMH Medium model running on Nvidia P100 GPU takes around 2 minutes per sample). Thus we obtain 8 sets of 2500 samples, four for each author.

Setup 2 for BERT evaluations

Datasets for the visual and BLEU evaluations created in Setup 1 have an insufficient number of samples per checkpoint (only 100) to be used for training the BERT-based classifiers. For this reason, we create a separate set of 8 datasets by fine-tuning all of our GPT-2 models on both datasets for 10K steps and generate 1K samples at each 1K steps checkpoint. We have chosen this span of checkpoints because it covers the sweet spot where the evaluation error is at the lowest, and we can observe the quality of the samples immediately before and after that point. Thus we obtain 8 datasets of 10K samples. The samples are limited to 600 tokens, since as we explain later, we use only the first 20 lines of each sample in Experiments 3 and 4.

Part 1—Evaluation of the overfitted models

In this section, we analyse the impact of overfitting on the quality of the produced samples. This is to emphasize the importance of early stopping of the fine-tuning process and to explore how the existing quantitative metrics (such as BLEU) correlate with the overfitting of GPT-2.

Experiment 1—Visual evaluation of text quality

During this research, we have observed that while generating text with the full length possible, many samples come with significant errors. We have decided to establish whether there is any regularity in the production of the malformed samples. For this purpose, we analyse the datasets from Setup 1. From every 100 samples generated at a specific checkpoint, 10 samples are selected at random and evaluated manually by the authors using the following procedure: when looking at the sample, we check at which point within the sample the text becomes unintelligible or contains obvious errors. We take note of the line number where this happens, and we take note of the total number of lines in the sample (including the blank lines at the end). Then we calculate the percentage of the correctly produced lines. After that, we calculate the average value for those 10 samples. We repeat this for all 25 checkpoints. The results for both datasets are shown in Figure 2. Examples of correct and malformed samples generated in our experiments are available on our repository.

We can see that the Regular models score almost 100% across the whole range of fine-tuning checkpoints. For the LMH models, the percentage of correctly generated text within a sample is at its best at 10K and 20K checkpoints, and after that, it rapidly decreases to around 35% for the remaining checkpoints.

We have observed that once the errors start appearing in the samples generated by the LMH models, the remainder of the sample is almost always malformed. In contrast, the Regular models occasionally produce a few malformed lines in the middle of the sample, but the subsequent text is consistent again. The LMH models' output does not have this "self-repairing" property. We are aware that these results could be different if much larger or much smaller datasets were used for fine-tuning. The reason we chose datasets of this size is because of our objective of style preservation of an individual poet.

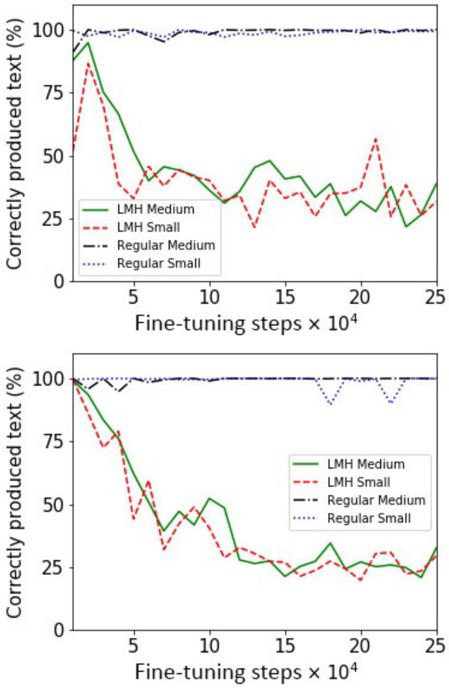


Figure 2: Results of the visual evaluation of text quality for samples generated from the Byron (top) and Shelley (bottom) datasets.

A well-formed sample with a Byron-style text of 1000 tokens usually spans around 45 to 80 lines in a text file. However, the malformed samples from the LMH models could sometimes exceed 180 lines, of which often only around 30 lines at the top of the sample are of good quality. With the Shelley-style samples, the malformed samples can exceed 250 lines, with more or less the same proportion of correctly produced text. This is because the number of tokens per line plummets, and many blank lines are inserted into the sample. This "stretching" phenomenon was not observed in the samples produced by the Regular models. One could raise a question whether these results were caused by poor data pre-processing. This, however, is unlikely since our data pre-processing was rigorous and comprehensive, as described in the section on Data Preparation. Given the quality and fineness of data used for fine tuning, it is clear that the errors in the samples must be caused by the properties of the models themselves.

Because of the malformed text in long samples, in the subsequent experiments with the BERT-based classifiers, we will limit the sample length to the first 20 lines of text from each sample. This is because our goal in the second part of this study is to evaluate only well-formed outputs, instead of learning to spot obvious errors, like repeated lines, garbled and unintelligible text, etc.

The results of the visual evaluation in Figure 2 show a deficiency of the LMH models after 10K-20K training steps with these specific datasets. When these results are contrasted with Figure 1, one can notice a clear correlation between overfitting—quantified by a high evaluation loss in Figure 1–and the ratio of malformed lines in Figure 2. Such a correlation in not present in the results of the Regular models, however, and their results in Figure 2 do not detect any malformed text according to the visual evaluation. For this reason, we perform a BLEU evaluation in the next experiment to see if the effect of overfitting can be uncovered in the samples from the Regular models.

Experiment 2—What does the BLEU evaluation tell us?

The visual evaluation has informed us that samples from the Regular models appear to be correctly produced across all the checkpoints from 10K to 250K fine-tuning steps, regardless of the increasing evaluation loss between those checkpoints. But, are there any noticeable and measurable changes in text quality between those checkpoints? In order to establish that, we perform a BLEU evaluation of those samples against the original dataset.

Bilingual Evaluation Understudy (BLEU) was originally designed to evaluate machine translation (Papineni et al. 2002), where we have the master translation of the sentence in the source language, and a candidate translation produced by the system. BLEU compares and counts the matching n-grams between the master and the candidate, resulting in a score between 0 and 1. The closer the score is to 1, the better for the translation task because this indicates that the candidate translation is more similar to the master translation. While not designed for that, BLEU is sometimes used to evaluate text quality (Yu et al. 2017), but when used for this purpose, it suffers from several deficiencies. Our objective, however, is to measure only the n-gram based similarity between the samples and the original text. We therefore expect that BLEU is an appropriate algorithm for our application because we have two types of text to compare, albeit we interpret the scores differently. Unlike in the translation tasks, in our research, we are aiming at a lower score, which indicates that fewer n-grams in the sample were copied from the original dataset, thus demonstrating a higher level of originality. In other words, we treat the BLEU score as a plagiarism detector on the n-gram level, which would be quantified by a high score. We use it to explore if there are any trends in the averaged BLEU scores between consecutive checkpoints. In our application of BLEU, we score each sample against the original dataset, and then average the results, similarly to what was proposed by (Yu et al. 2017) and implemented by (Wesek 2019). The implementation of BLEU used in our code was from the NLTK library[2].

Like in Experiment 1, we follow Setup 1, and we use 100 samples for each checkpoint from 10K to 250K. Samples have the length of 1000 tokens. We compute BLEU for all the 100 samples at each specific checkpoint, and we take the mean of those values to obtain a single value per checkpoint. Figure 3 shows that the BLEU score consistently increases with the model fine-tuning steps for both Regular models. This indicates an increasing similarity of the samples compared to the original dataset, when the models are fine-tuned for longer. This increasing BLEU score basically means that GPT-2 plagiarizes n-grams from the original dataset. On the other hand, the BLEU scores for samples from the LMH models do not increase in the same way. This is because of the increasingly high amount of the malformed text, which prevents the BLEU score from rising. When we evaluated the samples truncated to 200 tokens (which discarded all the malformed text in the samples from the LMH models), the increase of the BLEU scores for both types of models was very similar. In other words, they consistently rose with the number of fine-tuning steps.

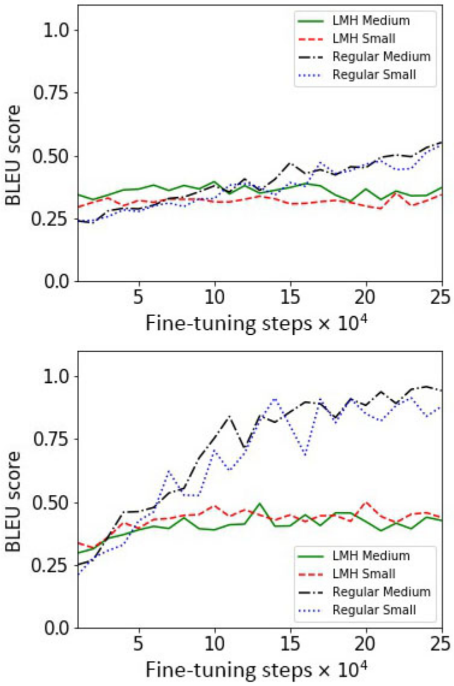


Figure 3: BLEU scores for Byron (top) and Shelley (bottom) calculated for samples with 1000 tokens length.

Altogether, we have observed that overfitting in the LMH models is easy to spot because the samples are malformed in an obvious way, but we must remain cautious about overfitting the Regular models, where the results of overfitting are not noticeable by our visual evaluation (Figure 2). Furthermore, the samples from the Regular models appear to be well-formed across all the checkpoints, while containing increasingly higher levels of plagiarized text or n-grams. This means that both automated and human evaluators could be misled by such outputs, since we cannot expect them to have memorized the original text. As a result, it is advisable to stop the fine-tuning process when the evaluation loss is

[2] Natural Language Toolkit Documentation:
https://www.nltk.org/

minimised or soon after when the samples start to be well produced after the initial learning of the model. Excessive training can lead to plagiarised n-grams, with even entire blocks of text repeated in GPT-2's output for the numbers of fine-tuning steps above 100K on our datasets.

A base requirement for GPT-2 to be creative is that it creates semantically and syntactically correct output. However, one way for it to do so is to just copy the source material, so high BLEU scores (or another measure for plagiarism detection) can indicate that the system is moving away from the novelty, which is also fundamentally necessary for creativity. As such, using a system like BLEU can be helpful in experiments with GPT-2-based poetry.

Part 2—BERT evaluation of correctly produced samples

In this section, we perform two experiments with a BERT-based binary classifier to establish which of the four GPT-2 models used is best at replicating the authors' style.

Experiment 3—Can fine-tuned GPT-2 outwit BERT?

In this experiment, we aim to verify if GPT-2's outputs can be of a sufficiently high quality to confuse a BERT-based classifier trained to distinguish the generated text from the original work of a poet. The experiments in the previous section have warned us about the plagiarized text (n-grams) in the samples from the Regular models. This behaviour, however, starts to become prominent late into the fine-tuning process, which, in our case, is well after 10K fine-tuning steps. For this reason, the samples we evaluate in this experiment are produced from checkpoints at 1K to 10K fine-tuning steps, which is before the plagiarism starts having significant impact.

Our previous experiments with the visual and BLEU evaluation have informed us that in the samples from the LMH models, for most checkpoints, only the first part is of high quality, and therefore, in this experiment, we use only the first 20 lines of each sample. This is because our intention is to apply BERT to the text that is correctly produced, and not learn to spot obvious mistakes, like in the malformed samples. The original text of the author is also split into 20-line fragments before it is fed into BERT, both for learning the classifier and for prediction. For each BERT-classifier, we prepared a dataset of 1K samples for each label, giving 2K samples in total, where label 0 is for samples from the original dataset, and label 1 for the samples generated by GPT-2. The train/test/validation split is 70/25/5. We use "bert-base-uncased" from the Transformers library, which is trained for 20 epochs, with the Adam optimizer, and a learning rate of 2e-5. The average classification accuracy on test data is taken as a final result of classification. Since our classification problem is balanced, i.e. the input contains the same amount of samples for both labels, we do not need to calculate Precision, Recall and F1 scores, and we can rely solely on accuracy. As described in the section on data preparation, we train the GPT-2 models for 10K steps and generate 1K samples at each 1K steps interval. This scope of checkpoints encompasses the best evaluation loss for both types of models (Figure 1). This allows us to observe the changes in classification results for 10 checkpoints, with a separate dataset of 2×1K samples for each checkpoint. Thus we train ten BERT-classifiers for each dataset/GPT-2 model pair.

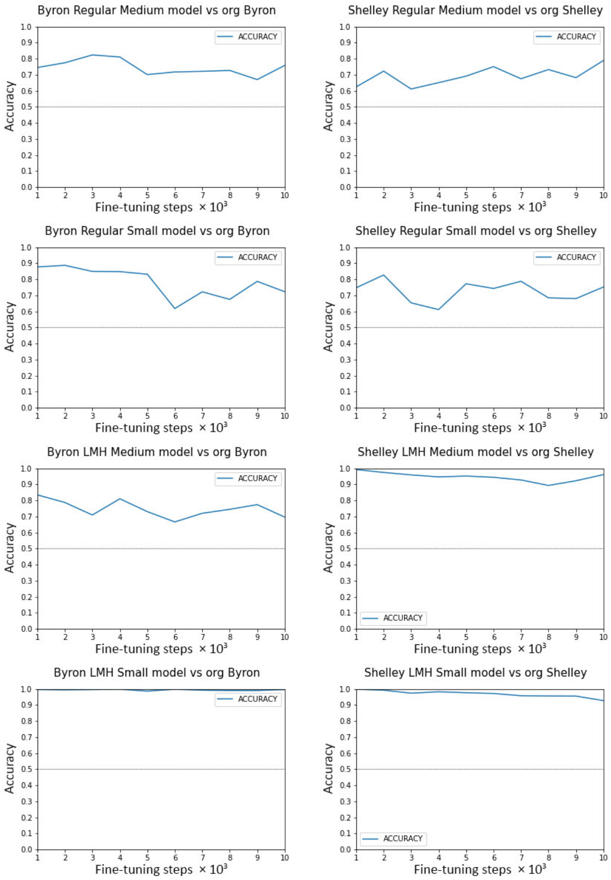


Figure 4: Results of the BERT evaluation of the Byron (left) and Shelley (right) checkpoint samples produced by the four different GPT-2 models. Row 1 (top): Regular Medium models, Row 2: Regular Small, Row 3: LMH Medium, Row 4 (bottom): LMH Small.

It is important to note how we interpret the results. In most applications, the closer the values of accuracy are to 1, the more desirable the performance of the classifier is. In this experiment, however, the GPT-2's output will be deemed to be indistinguishable from the original text, when the BERT classifier behaves like a random classifier. We know that a random classifier has an expected accuracy of 0.5 in a two-class problem. On the other end, an accuracy of 0 would mean the model still distinguishes between classes. For these reasons, the accuracy of 0.5 is our desirable target that can be seen as the evidence of GPT-2's high performance in generating high quality text. We can think of this as a sort of a "Turing test", which is successful when the BERT classifier behaves like a random classifier. This adversarial evaluation approach has been used before in NLP with both human (Köbis and Mossink 2021) and automated evaluations (Bowman et al. 2015).

Figure 4 shows the classification results for Byron and Shelley's datasets generated from all four GPT-2 models used in this study. The results show that the Regular models perform well on all checkpoints on both datasets, but interestingly the Regular Small model required 6K steps to reach its optimal performance, while its lowest evaluation loss is at 1700 steps (Figure 1). This indicates that we cannot rely on the evaluation loss alone, but instead, we may want to analyse the models' output to establish the optimal early stopping time. The LMH Medium model performs well on Byron, but very poorly on Shelley. The LMH Small models have the lowest scores on both datasets. Thus, the Regular models appear to be a more reliable choice.

All models appear to have similar, stable performance across all 10 checkpoints (Figure 4), and thus these results do not correlate with the evaluation loss. This is because they are for the numbers of fine-tuning steps when no strong overfitting is observed (Figure 1) and the BLEU scores did not increase significantly yet (Figure 3).

In the next experiment, we use BERT and Setup 2 again to compare the GPT-2 implementations, but using a different experimental design.

Experiment 4—Which GPT-2 is better at replicating the author?

In the previous experiment, we were classifying samples from the original dataset (label 0) against samples from a specific GPT-2 model (label 1). This gave us some indication as to which model is better at replicating the authors' style.

Here, we propose a novel setup for text evaluation with the BERT-based classifier. This time we take samples from two different GPT-2 models, which we assign labels 0 and 1, and we classify against them only the samples from the original author's writing. The accuracy is averaged, and it indicates to which models' output the samples from the original are closer.

To train the classifier, we use the dataset of 1K samples generated from two different GPT-2 models in a given pair after 10K steps of training. We selected this number of fine-tuning step because (according to our previous experiments) both the evaluation loss (Figure 1) and the BLEU scores (Figure 3) show that we are not comparing overfitted models that plagiarize the original works. As explained before, only the first 20 lines of each sample are used in this experiment (see Setup 2). Just like in the previous experiment, we use "bert-base-uncased" from the Transformers library, which is trained for 20 epochs, with the Adam optimizer, and a learning rate of 2e-5. Every classifier is tested on an additional test dataset of 1K samples randomly selected from the original authors' corpus, each sample 20 lines in length. The results are averaged, giving a single value of accuracy. This value indicates which label the original dataset is closer to, i.e., which GPT-2 generates text more similar to the original work. Since we have four different models, we can create six possible pairs (Table 2 and Table 3) for each dataset.

Since the class labels are 0 and 1, Tables 2 and 3 can be interpreted in the following manner: when the score is smaller than 0.5, then the model listed in the left column wins, and conversely, when the score is greater than 0.5, then the model in the right column is the winner.

Tables 2 and 3 show that the Regular Medium model wins on both datasets. On the Byron dataset, the Regular Medium model is clearly the best, Regular Small and LMH Medium are both second best and appear to have very similar performance, while LMH Small scores the lowest. This is consistent with the findings from the previous experiment in which the Regular models led to better results. Regarding the Shelley dataset, the Regular Medium model again performs the best, but the other three models have similar performance. This could indicate that the LMH Small model performs better on the Shelley dataset because it is much smaller than the Byron dataset.

Label 0	Label 1	Score
Regular Medium	LMH Medium	0.32
Regular Small	LMH Small	0.08
Regular Medium	Regular Small	0.32
LMH Medium	LMH Small	0.09
Regular Small	LMH Medium	0.51
Regular Medium	LMH Small	0.08

Table 2: Results of Experiment 4. Byron's work is classified using BERT models trained on two types of GPT-2 generated data.

Label 0	Label 1	Score
Regular Medium	LMH Medium	0.31
Regular Small	LMH Small	0.49
Regular Medium	Regular Small	0.28
LMH Medium	LMH Small	0.54
Regular Small	LMH Medium	0.49
Regular Medium	LMH Small	0.31

Table 3: Results of Experiment 4. Shelley's work is classified using BERT models trained on two types of GPT-2 generated data.

To conclude this section, both evaluations with the BERT classifier—the first being a sort of a "Turing test", and the second being our novel setup—show that the Regular (OpenAI original release) models perform better in general. While we have to watch out for Regular models' tendency to plagiarize text, they could be a preferred choice, especially if we want to generate text with the full sample length of 1024 tokens.

Discussion

This pilot study represents initial explorations into investigating GPT-2 from a computational creativity perspective. The question of whether GPT-2 can generate high-quality poetry (Lamb, Brown, and Clarke 2016), or creative poetry (not necessarily the same goal, as reflected in (Jordanous 2018)), is much larger than the scope of this pilot study; here we focus on the initial steps of model selection for this domain and avoiding problems caused by and analysing consequences of overfitting. One critique of a system based on generating the style of a known poet (Gervás 2011) is how a poet's style can diverge during their career. This criticism deserves focused attention in our future work; it is not a straightforward question to address and will benefit much from collaboration with experts in English literature. A quick attempt to solve this problem might involve training the model by tagging the dataset with indicators of which period of the author's writing the specific parts come from, and then applying those tags during generation of poems.

A key question here is: is GPT-2 creative? Our work above does not answer that question but gives us some material to consider, which we structure according to the 'Four Ps' of creativity: Producer, Product, Process, Press (Jordanous 2016). GPT-2 can produce Products that might be deemed creative, and it learns from a controlled Press (environment) in the form of the input corpus (though it is not able to interact more widely with its Press). The Process is (very) arguably creative as an example of Boden's exploratory creativity (Boden 2004). Does GPT-2 possess attributes of a creative Person though? This is hard to claim; GPT was developed as a language model, not an AI system, and behaves as a tool to assist a user. That being said: we see such potential to enhance future GPT-x systems through computational creativity research, to make GPT more creative in its own right.

A related question is: is the core task of generating new poems in an existing author's style a valid computational creativity task. Brown and Jordanous consider exactly this question in a new paper (Brown and Jordanous 2022), and give an overall fairly positive answer; in particular, the questions we are addressing in this paper (in particular around avoiding plagiarism and around ensuring high-quality outputs) provide some evidence that the task we are addressing is non-trivial in important ways, and hence more likely to require proper computational creativity effort.

Conclusion

In this study, we analysed the GPT-2 models' outputs with a twofold objective: 1) to observe how overfitting affects the output of GPT-2, and 2) to compare the quality of the output from various versions of GPT-2.

While working with deep neural networks, we are normally looking for the point of the lowest evaluation loss because this is known to lead to the best generalisation (Shalev-Shwartz and Ben-David 2014), though we also know (Domingos 1998) that a bit of overtraining or more complexity can lead to better results on a specific test dataset. The lowest evaluation loss in our results happens very early in the fine-tuning process, that is, before 10K steps in each case. We have trained our models for much longer (up to 250K steps) in order to observe how overfitting affects the quality of the generated samples. In the case of the LMH models, overfitting manifests by production of malformed samples. In the case of the Regular models, the samples are almost always well formed, even for 250K training steps. However, using the BLEU evaluation, we have discovered that, with overfitting, the BLEU score of the samples evaluated against the original dataset is continuously rising, which means that samples contain higher and higher levels of n-grams plagiarized from the original corpus. Effectively, the samples are becoming a kind of collage or pastiche of the original, instead of being fluently created. Such samples could easily mislead both human and automated judges and make them believe that the samples are "good", while they simply contain text plagiarized from the original sources. We should add that with extreme overfitting observed after around 100K training steps, our GPT-2 models plagiarize even long blocks of text (e.g. 50 lines of the original poem can be reproduced by a GPT-2 in many runs; see our supplementary repository for specific examples). Overall, given that we know that machine learning researchers recommend more complex models (Domingos 1998), we advice to stop the fine-tuning process as soon as the samples start looking "good" after the initial learning of the model, and to always check for plagiarism, which can mislead metrics and evaluation that cannot flag plagiarism.

With regards to the second objective of automated evaluation of samples to determine which GPT-2 model would be preferred for poetry generation, we have used two different setups of the BERT-based binary classifier. The first experiment with BERT showed that Regular models are a more reliable choice.

The second BERT experiment, which is our novel approach, in which we classify the samples from the original dataset by the classifier trained on samples from two different GPT-2 models, shows a clear advantage of the Regular Medium model on both datasets. These results are consistent with those of the first setup, and confirm the above findings that the Regular models appear to perform better than LMH models on style preservation tasks.

This study stresses the importance of applying various methods of text evaluation. As of yet, we do not have a single method that would tell us which text is "better". Quantitative methods, like BLEU, can tell us about the repeated n-grams, but they do not inform us about the creative quality of the text. Deep neural network classifiers offer a viable solution, but they can be misled by plagiarised outputs that would be indistinguishable from the original data. Based on our findings, we advice to always use multiple methods of evaluation.

The evaluation setups that we investigated require further research. Will they still provide valid insights when applied to Large and XLarge GPT-2 models, or even the larger models like GPT-3 or EleutherAI's GPT models? Will they be different when applied to much larger or much smaller fine-tuning datasets? If different versions of BERT were used, would they produce different evaluations? This is ongoing research and we hope this study has offered useful insights into the practicalities of evaluating GPT-2-produced text.

The overall contribution of this paper could be seen as both to the AI tools for computational creativity and to the methodologies, which seem to be quite intricate given that the machine learning models lose their generalization capability when overfitted, and can, therefore, plagiarise easily.

Author Contributions

Experimental design: PS with MG, AJ, DB, MP; experimental implementation: PS; writing: PS with MG, AJ, DB, MP, editing: MG, AJ, DB, MP.

Acknowledgements

We thank the anonymous reviewers for their feedback and suggested references. The work of DB is supported by a Discovery Grant from the Natural Sciences and Engineering Council of Canada. MP is supported by the University of Kent GTA Studentship Award, Prins Bernhard Cultuurfonds, Hendrik Mullerfonds, and Vreedefonds.

References

Alvarez Cos, M.; Perez y Perez, R.; and Aliseda, A. 2007. A generative grammar for pre-hispanic production: The case of El Tajin style. In *Proceedings of the 4th International Joint Workshop on Computational Creativity*, 39–46.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Black, S.; Biderman, S.; Hallahan, E.; Anthony, Q.; Gao, L.; Golding, L.; He, H.; Leahy, C.; McDonell, K.; Phang, J.; et al. 2022. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*.

Boden, M. A. 2004. *The creative mind: Myths and mechanisms*. Routledge.

Bowman, S. R.; Vilnis, L.; Vinyals, O.; Dai, A. M.; Jozefowicz, R.; and Bengio, S. 2015. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*.

Brown, D., and Jordanous, A. 2022. Is style reproduction a computational creativity task? In *Proceedings of the 13th International Conference on Computational Creativity*.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Corneli, J.; Jordanous, A.; Shepperd, R.; Llano, M. T.; Misztal, J.; Colton, S.; and Guckelsberger, C. 2015. Computational poetry workshop: Making sense of work in progress.

Dale, R. 2021. GPT-3: What's it good for? *Natural Language Engineering* 27(1):113–118.

Das, A., and Gambäck, B. 2014. Poetic machine: Computational creativity for automatic poetry generation in Bengali. In *ICCC*, 230–238.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Domingos, P. 1998. Occam's two razors: the sharp and the blunt. In *KDD*, 37–43.

Elgammal, A.; Liu, B.; Elhoseiny, M.; and Mazzone, M. 2017. CAN: Creative adversarial networks, generating "Art" by learning about styles and deviating from style norms. *arXiv preprint arXiv:1706.07068*.

Ens, J., and Pasquier, P. 2018. Caemsi: A cross-domain analytic evaluation methodology for style imitation. In *ICCC*, 64–71.

Falk, M. 2021. Artificial stupidity. *Interdisciplinary Science Reviews* 46(1-2):36–52.

Floridi, L., and Chiriatti, M. 2020. GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines* 30(4):681–694.

Gervás, P. 2011. Dynamic inspiring sets for sustained novelty in poetry generation. In *ICCC*, 111–116.

Hämäläinen, M. 2018. Harnessing NLG to create Finnish poetry automatically. In *Proceedings of the ninth international conference on computational creativity*. Association for Computational Creativity (ACC).

Jordanous, A. 2016. Four pppperspectives on computational creativity in theory and in practice. *Connection Science* 28(2):194–216.

Jordanous, A. 2018. Creativity vs quality: why the distinction matters when evaluating computational creativity systems. AISB.

Köbis, N., and Mossink, L. D. 2021. Artificial intelligence versus Maya Angelou: Experimental evidence that people cannot differentiate AI-generated from human-written poetry. *Computers in human behavior* 114:106553.

Lamb, C., and Brown, D. G. 2019. Twitsong 3.0: Towards semantic revisions in computational poetry. In *ICCC*, 212–219.

Lamb, C.; Brown, D. G.; and Clarke, C. 2016. Evaluating digital poetry: Insights from the CAT. In *Proceedings of the seventh international conference on computational creativity*.

Lamb, C.; Brown, D. G.; and Clarke, C. L. 2017. A taxonomy of generative poetry techniques. *Journal of Mathematics and the Arts* 11(3):159–179.

Lee, J.-S., and Hsiang, J. 2020a. Patent claim generation by fine-tuning openai GPT-2. *World Patent Information* 62:101983.

Lee, J.-S., and Hsiang, J. 2020b. PatentTransformer-2: Controlling patent text generation by structural metadata. *arXiv preprint arXiv:2001.03708*.

Lee, J.-S. 2019. Personalized patent claim generation and measurement. *arXiv preprint arXiv:1912.03502*.

Li, P.; Zhang, H.; Liu, X.; and Shi, S. 2020. Songnet: Rigid formats controlled text generation. *arXiv preprint arXiv:2004.08022*.

Liao, Y.; Wang, Y.; Liu, Q.; and Jiang, X. 2019. GPT-based generation for classical chinese poetry. *arXiv preprint arXiv:1907.00151*.

Lieber, O.; Sharir, O.; Lenz, B.; and Shoham, Y. 2021. Jurassic-1: Technical details and evaluation. *White Paper. AI21 Labs*.

Loller-Andersen, M., and Gambäck, B. 2018. Deep learning-based poetry generation given visual input. In *ICCC*, 240–247.

Manurung, H. 2004. An evolutionary algorithm approach to poetry generation.

Misztal, J., and Indurkhya, B. 2014. Poetry generation system with an emotional personality. In *ICCC*, 72–81.

Nikolov, N. I.; Malmi, E.; Northcutt, C. G.; and Parisi, L. 2020. Rapformer: Conditional rap lyrics generation with denoising autoencoders. *arXiv preprint arXiv:2004.03965*.

Oliveira, H. G., and Cardoso, A. 2015. Poetry generation with PoeTryMe. In *Computational Creativity Research: Towards Creative Machines*. Springer. 243–266.

Oliveira, H. 2009. Automatic generation of poetry: an overview. *Universidade de Coimbra*.

Oliveira, H. G. 2017. A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. In *Proceedings of the 10th international conference on natural language generation*, 11–20.

Oliveira, H. G. 2021. Exploring a masked language model for creative text transformation. 62–71.

OpenAI. 2021. openai/gpt-2: Code for the paper ”language models are unsupervised multitask learners”.

Pachet, F., and Roy, P. 2014. Non-conformant harmonization: the real book in the style of take 6. In *ICCC*, 100–107.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.

Project Gutenberg. 2020. `http://gutenberg.org/`.

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding by generative pre-training.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1(8):9.

Rae, J. W.; Borgeaud, S.; Cai, T.; Millican, K.; Hoffmann, J.; Song, F.; Aslanides, J.; Henderson, S.; Ring, R.; Young, S.; et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

Rahman, F., and Manurung, R. 2011. Multiobjective optimization for meaningful metrical poetry. In *ICCC*, 4–9.

Rashel, F., and Manurung, R. 2014. Pemuisi: a constraint satisfaction-based generator of topical Indonesian poetry. In *ICCC*, 82–90.

Shalev-Shwartz, S., and Ben-David, S. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge University Press.

Shepperd, N. 2021. nshepperd/gpt-2: Code for the paper “Language Models are Unsupervised Multitask Learners”.

Smith, S.; Patwary, M.; Norick, B.; LeGresley, P.; Rajbhandari, S.; Casper, J.; Liu, Z.; Prabhumoye, S.; Zerveas, G.; Korthikanti, V.; et al. 2022. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, 3104–3112.

Toivanen, J.; Järvisalo, M.; Toivonen, H.; et al. 2013. Harnessing constraint programming for poetry composition. In *The Fourth International Conference on Computational Creativity*. The University of Sydney.

Toivanen, J.; Gross, O.; Toivonen, H.; et al. 2014. “The Officer Is Taller Than You, Who Race Yourself!”: Using Document Specific Word Associations in Poetry Generation. In *Proceedings of the Fifth International Conference on Computational Creativity*. Jožef Stefan Institute.

Transformers Documentation. 2019. OpenAI GPT2 — transformers 4.5.0.dev0 documentation.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Veale, T. 2013. Less rhyme, more reason: Knowledge-based poetry generation with feeling, insight and wit. In *ICCC*, 152–159.

Ventura, D. 2016. Mere generation: Essential barometer or dated concept. In *Proceedings of the Seventh International Conference on Computational Creativity*, 17–24. Sony CSL, Paris.

Wang, B., and Komatsuzaki, A. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Wesek, A. 2019. A comprehensive study of state-of-the-art word embedding algorithms for natural language generation. `https://www.cs.kent.ac.uk/people/staff/mg483/documents/wesek19lyrics.pdf`. University of Kent, Unpublished MSc Thesis.

Wöckener, J.; Haider, T.; Miller, T.; Nguyen, T. T. L.; Pham, M. V.; Belouadi, J.; Eger, S.; et al. 2021. End-to-end style-conditioned poetry generation: What does it take to learn from examples alone? In *Proceedings of the 5th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, 57–66.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Yu, L.; Zhang, W.; Wang, J.; and Yu, Y. 2017. SeqGAN: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Spinning Coherent Interactive Fiction through Foundation Model Prompts

Alex Calderwood, Noah Wardrip-Fruin, Michael Mateas
Expressive Intelligence Studio
Computational Media Department
University of California, Santa Cruz
alexcwd,nwardrip,mmateas@ucsc.edu

Abstract

We present *Spindle*, a mixed initiative tool for authoring choice-based interactive fiction that targets *Twine*, a popular framework for text-based storygames. *Twine* artifacts have properties of both stories and games, placing our system at the intersection of Automated Game Design (AGD) and Automated Story Generation (ASG). We construct a generative pipeline that involves condensing narrative context into a compact representation in order to feed to a pretrained language model, which we further fine-tune. We demonstrate that, by maintaining narrative context in the prompt presented to the language model, we can greatly improve over the loss of long-term coherence that still plagues such models. Our story compression technique for representing narrative context uses a handful of freely available natural language processing libraries and models, demonstrating that such interpretive pipelines can be built with limited computational resources and low cost. The resulting tool is capable of producing full-text branching narratives, or of generating individual passages that maintain a high degree of narrative coherence with the prior passages. The framework we design is both language model-agnostic and narrative theory agnostic, allowing future researchers to easily expand on it with new language models and story representations. We release our code under the BSD-4-Clause[1].

Introduction

While large language models have proven capable of producing highly coherent text for such purposes as autocompletion and chat bots, less research effort has gone into exploring the entirely new forms of media they enable. We are interested in ludic spaces that may be created by advancements in language model-based story generation, and present one such tool, a system for authoring interactive fiction which allows for expressive interaction with an AI model. By focusing on the experience of writing as a target for AI augmentation as well as a source of entertainment, this project is influenced by two fields of study: computer mediated writing and automated game design, resulting in a system that is both engaging and encouraging of narrative experimentation.

Our tool allows users to author interactive-fiction narratives with the Twine framework, alternately 'passing the keyboard' between the system's language model and the user at will. When queried, the system utilizes a method of narrative compression to come up with an understanding of the current thrust of the story, and uses that understanding to condition the generation of the next passage. This approach is novel in how it uses the compressed narrative context to improve the coherence of generated narratives through prompt engineering. Prompt engineering is a new and understudied paradigm where pretrained language models are guided towards better responses by providing a succinct (often templated) prompt to the model at the time of prediction (Reynolds and McDonell 2021). This method of iterating on a series of prompts has been successfully used in the computational creativity literature for text to image generation (Liu and Chilton 2021).

Unlike many past story generation techniques, which generate a list of ordered plot events, our system generates fully realized multi-passage branching narratives, with a capability for mixed-initiative use. The generated passages follow Twine syntax and include outgoing links (player decision points) and corresponding passage titles. Generated passages display a level of narrative coherence that allows the model to 'yes-and' the user's apparent authorial intention, while still enabling a degree of defamiliarization that results from the composition of nearly appropriate text, an attribute of AI writing which has been said to be prized by writers including the novelists Robin Sloan and Sigal Samuel, who respectively envision an AI writing assistant as "less Clippy, more séance" and describe feeling "strangely moved" by AI writing (Calderwood et al. 2020).

We fine-tune GPT-3, a model notable for its introduction of meta-learning, or zero-shot learning, to language generation (Brown et al. 2020). Under the zero-shot paradigm, queries consist of a small number of example (*prompt*, *response*) pairs. Further, fine-tuning such a model allows it to capture the grammar and stylistic accompaniments of a structured corpus such as our dataset of Twine stories. The model has notably been used to enable gameplay in *AI Dungeon*[2], a text adventure game that allows arbitrary player input (Hua and Raley 2020). Accessing the model through the

[1] https://github.com/alex-calderwood/spindle

[2] https://play.aidungeon.io/

OpenAI API and utilizing other open-source NLP packages, we fine-tune our GPT-3 based models at a small fraction of the cost of traditional NLP development, and all systems easily run on an internet-connected laptop.

Background

Creating a Twine game is an act of writing as well as interaction design; Twine games are 'played' by reading and read through play, and generating these artifacts requires both story and game generation techniques.

Automated Story Generation

Automated story generation has long been considered a grand challenge of artificial intelligence. Early systems used symbolic reasoning (Meehan 1977; Lebowitz 1987), often utilizing hierarchical generation based on grammar rules which provide model interpretability, clearly identifiable design spaces, easy extensibility, and little or no input data, though they sometimes lack robustness (Black and Wilensky 1979; Riedl and Young 2010). Tracery has found popularity as an author-focused generative text tool with non-deterministic grammar production (Compton, Kybartas, and Mateas 2015). When it comes to full stories, "hand-authored templates are seen as insufficient for large scale narrative generation" (Roemmele 2018), though the combination of generative grammars with language modeling has not been sufficiently explored. A major component of many narrative generation systems includes a planner, a system that searches for a goal story state from the current state to find the series of narrative actions needed to bridge that gap (Porteous 2016; Young et al. 2013). StoryAssembler is a relevant example of a system that uses a planner to generate Twine-like interactive narratives (Garbe et al. 2019).

Utilizing statistical NLP, Chambers and Jurafsky (Chambers and Jurafsky 2008) define the *narrative cloze task* as the prediction of missing narrative events from an event sequence, which they produced from short plot descriptions. Their technique involves running a dependency parser over a collection of plot summaries to produce grammatical relationships, using these to extract a sequence of predicates along with their subject and object noun phrases, and then resolving these into ordered event chains. Pichota and Mooney (Pichotta and Mooney 2016) build on this framework, utilizing an LSTM architecture to extract (s, v, o, m) event tuples from Wikipedia articles (corresponding to subject, verb, object, and modifier, respectively). "John and Mary went to the store," becomes $(john, go, store, \emptyset)$. This event list may be thought of as a high-resolution, but potentially low accuracy, representation of the narratological notion of a fabula, or the chronological ordering of events in a narrative (Bal and Van Boheemen 2009).

Martin et al. follow this approach with a three part decoupled architecture: a system that extracts event tuples from unstructured narratives, an event2event model that predicts the next event predicated on the previous events, and an event2sentence model which expands a sequence of events back into natural language (Martin et al. 2018). To improve the model's performance, they used entity recognition to generalize characters and locations during prediction, which were memorized and later in-filled. This work serves as the closest inspiration for our approach. Our use of modern large-pretrained language models allows us to combine the event prediction and text generation steps into one text completion step, and our narrative interpretation module functions as a memory of important narrative entities without the need for back-filling generalized text.

Many attempts at narrative generation focus just on the production of language, but Mexica (Pérez and Sharples 2001) models creative writing as a cognitive process that consists of engaged and reflective states. In the reflective state, the model evaluates coherence, interestingness, and novelty of the generated sequences for text refinement. Our strategy utilizes reflexive interpretation to increase coherence of the proceeding passages. Like the generation of long-form text, automated interpretation of full stories has proven challenging, not least because it relies on systems such as long distance character co-reference resolution to be of a high-accuracy. BookNLP (Bamman, Underwood, and Smith 2014) is one of the few systems that approaches this tough problem. Increasing the quality of generated stories will likely need to incorporate global measures of coherence, and will therefore likely require the semantic readings that a system like BookNLP can provide.

The Virtual Storyteller System is a plot generation system that used a director model to orchestrate emotional episodic plots (Theune et al. 2003; Theune et al. 2004). The authors imagine a user acting as a character within the narrative, allowing them to influence the course of the story somewhat like a role playing game, not dissimilar to our mixed-initiative writer model.

Relation to Automated Game Design (AGD)

Full-game generation has historically focused on games that can be automatically analyzed with automated game players or a stack of static evaluation criterion (Pell 1992; Browne and Maire 2010; Cook, Colton, and Gow 2016). Such game generators seek to maximize a particular optimization function that guides search, which may be grammar-based, constraint-satisfying, or evolutionary in nature. They often utilize an intermediate Game Description Language (GDL) (Schaul 2013; Summerville et al. 2018; Duplantis et al. 2021), typically a text representation that is designed to have a high expressive range, produce interesting games a high proportion of the time, and be suitable for automatic search. Both Inform 7 and Ceptre can be viewed as narrative GDLs; the former is used in the TextWorld framework (Côté et al. 2018). In our case, Twine's Twee syntax[3] is modeled and generated analogously to these GDL's. From it, the Twee compiler targets interactive HTML.

The automated game players in AGD are intermediate processes that form interpretations of games, similar to the readings our system produces to conduct the narrative flow.

Cook et al. point out that most automatic game generation scholarship has gone towards "objectives, obstacles, and the notion of challenge" (Cook 2015). Game generators have

[3] Examples at: https://dan-q.github.io/twee2/tutorial.html

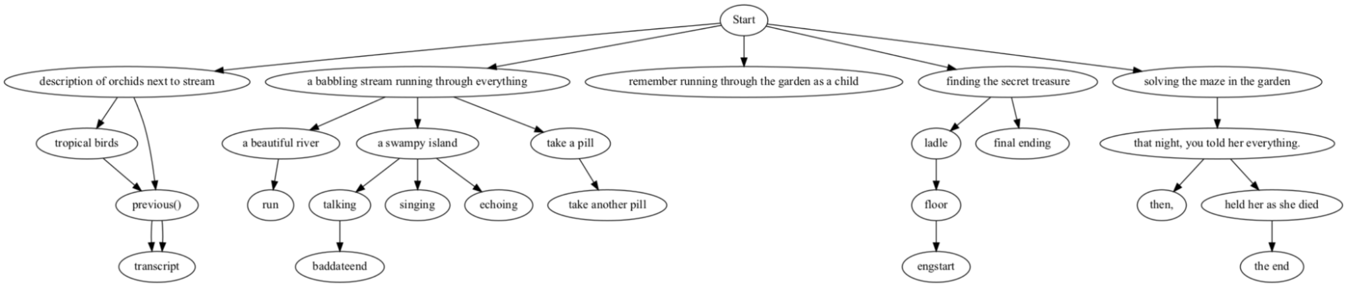

Start
description of orchids next to stream
a babbling stream running through everything
remember running through the garden as a child
finding the secret treasure
solving the maze in the garden
tropical birds
a beautiful river
a swampy island
take a pill
ladle
final ending
that night, you told her everything.
previous()
run
talking
singing
echoing
take another pill
floor
then,
held her as she died
transcript
baddateend
engstart
the end

Figure 1: A small Twine tree. The *Start* node (passage and links) was written by hand, further passages and titles were generated by *Spindle*. For space, only passage titles links are shown.

recently begun exploring the hard problem of automatically producing games with semantic, thematic, and cultural import, rather than focusing on fun or challenge. In (Cook 2015), the authors point to narrow readings of the word 'game' as hampering fruitful experimentation in generative systems. Unanswerable questions like "Is this a game?", "Is this a story?", or "Is this art?" often surface at challenging moments in the development of new expressive mediums, defensively boxing-in existing endeavours rather than nurturing interdisciplinary growth. Game-O-Matic is the first game generation system that attempts to reason about the rhetorical relationships between objects that make up the game, utilizing the theory of operational logics to build up an understanding of possible player interpretations (Treanor et al. 2012). Gemini takes this a step further, using answer set programming to build up a proceduralist reading of a game as it is generated, using that reading to influence generation (Summerville et al. 2018). In (Cook 2021) the authors attempted to generate games without scores, focusing on the aesthetic experience of play. It is worth noting that they found it challenging to determine their level of success or generate through a broad expressive range, in part due to the unclear notion of success.

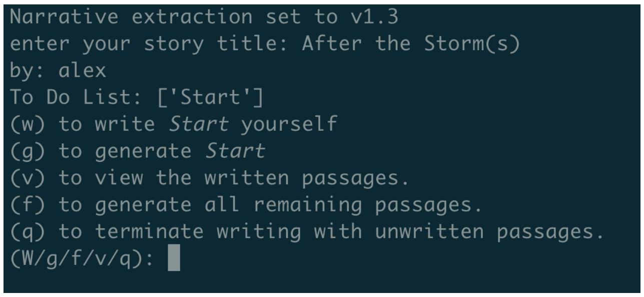
Automatic game generation and games as a whole have seen relatively slow adoption of language models. AI Dungeon is one notable exception (Hua and Raley 2020). A few reasons for this may be that language models are unpredictable, sometimes producing undesirable sexist, racist, and otherwise inappropriate language. Twine games are historically inclusive of explicit and otherwise non-normative content (Harvey 2014), meriting a conversation about what a nuanced treatment of complex social, sexual, and psychological issues looks like in the context of these models.

Additionally, getting these models to understand narrative context is difficult, as we will see. Some work has been done to use language models to evaluate text games (Côté et al. 2018; Kostka et al. 2017a; Kostka et al. 2017b). Fan et al. use neural generation to populate the world of a text-adventure game (Fan et al. 2020), but did not attempt full narrative generation. Earlier games such as Scribblenauts[4] have used human specified corpuses as safer and more reliable tools for injecting language understanding into games. (Barros et al. 2019) used Wikipedia as a corpus to automatically generate murder mystery games populated with characters and appropriately themed puzzles.

Twine

Twine is a platform designed to give non-coders the ability to author branching interactive fiction narratives. Such games are designed and written via a visual authoring tool that positions narrative passages on an interface resembling paper notes connected by strings on a corkboard (Friedhoff 2013). Passages are Twine's "equivalent of a page in a Choose Your Own Adventure book" (Friedhoff 2013), containing markdown-style hyperlinks to other passages. Gameplay consists simply in clicking between passage links, which may simulate making narrative choices for a first or third person character. Passages and links are handwritten, as contrasted with more open-ended commands available in parser games. Collections of Twine games include the Interactive Fiction Database[5] and itch.io[6].

Mixed-Initiative Interface



```
Narrative extraction set to v1.3
enter your story title: After the Storm(s)
by: alex
To Do List: ['Start']
(w) to write Start yourself
(g) to generate Start
(v) to view the written passages.
(f) to generate all remaining passages.
(q) to terminate writing with unwritten passages.
(W/g/f/v/q): 
```

Figure 2: Entry into the prototype interface.

Our system models writing a Twine story as constructing a graph of passages connected by their respective links, first soliciting a title and by-line from the author, and then moving into the authoring of the *Start* passage.

Spindle's interface is designed to treat both the human and machine as writers capable of drafting or refining passages

[4] https://en.wikipedia.org/wiki/Scribblenauts

[5] https://ifdb.org/viewlist?id=ax0yq2ksub57ie7o

[6] https://itch.io/games/made-with-twine

in the working story. The system maintains a ‘To Do List’, or a fringe of outgoing passage titles that have yet to be drafted.

At present, the system’s primary interaction loop begins by asking the user to select a passage title from the To Do List. Passage selection is accomplished with arrow keys — building a graphical UI is the focus of the next iteration of the tool. Next, the user is asked to indicate whether they would like to 1.) draft the passage body in a spartan text editor, 2.) indicate that the system should automatically generate the passage body using the narrative context stored for that passage, 3.) view the passages thus far written by the mixed-initiative system, 4.) generate N passages sequentially from the head of the list, or 5.) conclude the writing process, inserting placeholder text for unwritten passages.

The writer may for instance describe a branch point in an adventure game, say, a description of a cave with many tunnels for the player to explore. The tool is designed not to impose stylistic restrictions on the writer, so they may author link text in declarative or imperative styles (“You head into the entrance covered in stalactites” vs “Enter the slimy tunnel”) or any other text (“An unusual door”). Additionally, the writer can override the link text with specific titles for the linked passages so that “enter the cave” corresponds to the more conventionally formatted title “The Cave”.

The author may want to immediately write some passages and generate placeholder text for the others. Alternatively, they may decide to have the system generate all narrative branches recursively to create unplanned stories to play themselves. Using the tool in this manner allows it to produce ‘games that the developer wants to play’, which is commonly cited as a reason game makers get into development.

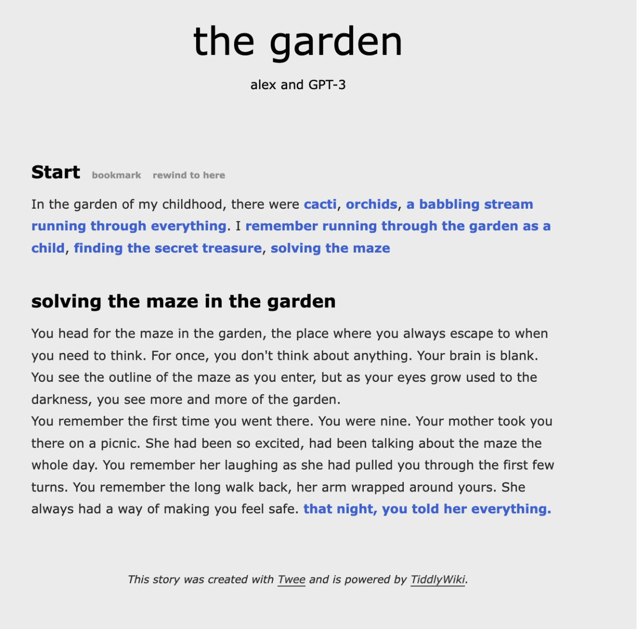
After a passage has been written by human or machine, it is parsed by the system to ensure it is well-formed Twine, and passed back to the writing protocol if not. In practice, the models we train rarely produce malformed syntax.

When all passages have been written or writing is manually concluded, the system uses the Twee compiler to convert the passages into playable HTML, saves those files, and then launches the Twine game in the default web browser as shown in Figure 3. At present, the games do not make use of custom visual styling, though we imagine future work will include conditionally generating stylesheets on the basis of a game’s thematic content.

Formulating the Generation Problem

Twine stories can be thought of as directed graphs composed of *passages* $p_i := (p_i^{title}, p_i^{body})$ connected to others by outgoing links. Each story contains a predefined $Start$ node. Any text in the body may be formatted as a link to a specified passage[7].

The model should ideally be aware of all preceding text in order to generate the next passage, according to the practice of automated feature engineering in which deep learning models themselves learn what information is relevant to their predictions. In practice, it has been observed that longer prompts decrease the coherence of completions in large language models and many strategies for increasing long form coherence have recently been proposed (Guan et al. 2021; Cho et al. 2018). Prompt engineering therefore necessitates a trade-off between including more information that might be useful to the prediction, and not ‘overwhelming’ the model with too much text. State of the art text completion models typically allow no more than around 1024 words shared between the input (prompt) and output (completion). So we define two auxiliary functions whose purpose is to condense preceding passages to a length that can be ingested without a loss of coherence due to to this phenomenon.



the garden

alex and GPT-3

Start bookmark rewind to here

In the garden of my childhood, there were **cacti**, **orchids**, **a babbling stream running through everything**. I **remember running through the garden as a child**, **finding the secret treasure**, **solving the maze**

solving the maze in the garden

You head for the maze in the garden, the place where you always escape to when you need to think. For once, you don't think about anything. Your brain is blank. You see the outline of the maze as you enter, but as your eyes grow used to the darkness, you see more and more of the garden.
You remember the first time you went there. You were nine. Your mother took you there on a picnic. She had been so excited, had been talking about the maze the whole day. You remember her laughing as she had pulled you through the first few turns. You remember the long walk back, her arm wrapped around yours. She always had a way of making you feel safe. **that night, you told her everything.**

This story was created with Twee and is powered by TiddlyWiki.

Figure 3: A screenshot of a playable Twine game produced by the system, corresponding to the tree in Figure 1

These abstract functions are implemented according to authorial goals and narratological propositions. In our experimentation, we repeatedly iterated over their practical implementation, each addition seeming to enhance narrative coherence along some dimension.

A *narrative reading* $R(p_i) = r_i$ is a representation of the features most salient to the given passage. These representations can be arbitrarily complex, holding on to important details such as characters and plot events, and stylistic features such as tone or point of view, throwing away details determined to be irrelevant.

Additionally, an *explanation* function $X(p_i) = X(r_0, ..., r_{i-1}) = \chi_{p_i}$ maps chains of narrative readings to a plain text description χ_{p_i} of the narrative context leading up to passage p_i. A sequence of individual readings $r_0, ..., r_{i-1}$ forms a context chain for a passage at depth i

[7] Advanced Twine authors sometimes make use of macros or custom Javascript code to allow conditional or stateful storytelling. While language models have shown an ability to produce complex code snippets, for simplicity we have excluded passages containing these features.

along the minimum spanning tree path between the passage and *Start*, p_0. The necessity of a plain text description falls out of the paradigm of transfer learning: using an out of domain model on a sub—or alternate—domain[8]. By transforming our latent story representations into English, we can plug in any large pre-trained model into the generation pipeline, and ensure our code base is largely model agnostic. Additionally, the narrative descriptions produced by this intermediate stage of the system are highly interpretable, as they are in plain English.

Text Preprocessing and Model Training

We use these X and R (implementations detailed in the next section) to approximate a distribution $\mathcal{L}$ over the narratives we aim to generate. We would like to provide the narrative context to the model, alongside the title of the passage to be generated: $\chi_p|p^{title}$ in order to produce a p^{body}. Fan et al. (Fan, Lewis, and Dauphin 2019) and others have shown that preprocessing text—injecting external knowledge, adding start and end tokens, and generalizing named entities—aids in generating coherent stories.

Data Processing

To begin, we convert Twine binaries into *Twee 2*, to represent the interactive stories textually as in Figure 4. Gathering sufficient Twee-formatted stories required modifying the Twee source[9] to handle decompilation from games natively authored in the graphical Twine interface.

We split our collection of Twine stories into passages, excluding passages with macros and non-unicode characters. Next, we run our narrative reader and explanation functions on our corpus to produce textual descriptions of the story up until this point. Finally, we build $(prompt, response)$ pairs for each passage by appending to the readings unique start and end tokens unlikely to appear in the corpus (e.g. $start$:=<||start||>):

$$prompt(p) = start|\chi_p|p^{title}|begin$$
$$response(p) = p^{body}|end$$

Of the 512 Twine stories downloaded from *itch.io*, only 82 were Twee decompilable, producing 10,784 passages with a total of 11,098 embedded links.

Training

For each of the following experiments, we feed the processed training examples to the OpenAI *completion* endpoint with the default meta-parameters: *top_p*, *temperature*, and *best_of* all set to 1. Experimenting with a different *top_p* would retrieve multiple responses from the model, which we could pick from based on some retrospective evaluation. This is left for future work, as is providing the user with temperature (stochasticity) control. Each experiment uses GPT-3's largest available 175B parameter *davinci* model, resulting in three separate fine-tuned models. Fine-tuning involves a limited retraining of a neural network's weights (until recently, this typically meant a selective retraining of only the latter levels of a feed-forward network) (Howard and Ruder 2018; Lester, Al-Rfou, and Constant 2021). Fine-tuning through the API abstracts the typical machine learning workflow, which typically requires splitting data into training and test sets, selecting an optimization function, and tuning hyperparameters.

Running a fine-tune job is currently free and took under three hours for each tune. The total cost incurred from using the models did not exceed $40, demonstrating that using open-source language toolkits on top of fine-tuning existing large language models is a viable way to cheaply develop creative language applications.

Narrative Reader Development

In this section we present a series of increasingly sophisticated implementations of the two interpretation functions and example passages generated by each from the same *Start* passage (Figure 4).

```
:: Start
A tornado devastated town. Now you, Robert,
    and Maya are alone.
[[Head outside|take them outside]]
[[Continue to shelter at home.|tell them to
    stay inside]]
```

Figure 4: Our example input passage setting up a story about a tornado. Bracketed text indicates link text followed by the passage linked. Double colons :: represent a passage title.

0. Naive Method

The first question we wanted to address was the capability of a fine-tuned *davinci* model to generate grammatical Twine stories, regardless of their narrative coherence. For this baseline, $X(p)$ returns no narrative context and passage bodies are generated solely based on their corresponding title.

We found that fine-tuning the *davinci* model with this narrative reader generated Twine stories with correct Twine syntax with high reliability despite the medium size of our tuning dataset. This freed us to refocus on the quality of the generated narratives in subsequent definitions of R and X. The passages are sometimes formatted with rich text that we were surprised to see render appropriately through Twee. Unsurprisingly, the passages followed English grammar and storytelling conventions.

The passages generated naively were disjointed; Figure 5 shows subsequent passages completely diverging from the tornado narrative. The only times any narrative continuity was observed between passages was when the titles were extremely explicit. Additionally, passages switch perspective and verb tense, introduce new characters and do not refer back to previous characters.

[8] https://ruder.io/state-of-transfer-learning-in-nlp/

[9] https://github.com/tweecode/twee

```
:: tell them to stay inside
 <div id="day">
Day 49
</div>
You are in your car. You’re on your way to
    pick up your daughter from daycare.
You’re still not ready to take this step.
    The thought of leaving your child alone
    for a whole day makes you cold.
You have a lot of work to do to finish your
    project. You need to finish the sound
    engine, implement a pathfinding system,
    design the world generation...
Your free time is running short.
You need to finish your project. You need to
     do it soon.
[[ok now im ready]]

:: ok now im ready
 You look at your opponent. His eyes are
     locked on you, a mixture of fear and
     anticipation on his face. You draw your
      sword and run towards him.
You both charge at each other, swinging your
     swords wildly. You manage to land a few
     hits, but he parries most of them.
    Suddenly, he jumps back and cowers in
    fear. "You can’t beat me", he cries. You
     look down and see the cause of his
    panic. You did a horrible job of
    cleaning the blood of your opponent off
    your sword. Horrible. You’ve been
    defeated by a sword-swinging mop.
[[You’re done.|engstart]]
```

Figure 5: Two passages generated via the naive method did not show continuity with the initial tornado passage (Figure 4) or each other, but demonstrate adherence to Twine syntax.

1. Zero-Shot Summarization

One concept for our narrative reader function was to use GPT-3 as a zero-shot summarizer, defining X and R according to Equations 1 and 2.

$$R(p_i) = ZeroShot(p_i) \tag{1}$$

$$X(r_0, ..., r_{i-1}) = r_0|...|r_{i-1} \tag{2}$$

Using $davinci$, our *ZeroShot* summarizer produced accurate single sentence summaries of complex multi-character passages using the following prompt scaffold:

```
My job is to summarize short stories
    into a single sentence. Here is an
    example of a longer story:
"""
<Passage Text>
"""
This is my one-sentence summary:
"""
```

While this prompt engineering approach to summarization was promising with *davinci*, we did not have the resources to perform the summarization to produce $(prompt, response)$ pairs on our full dataset. *Davinci* is 10x the cost of the next largest *curie* model, which tended to simply produce the first sentence of the passage rather than an accurate summarization. In order to fine-tune the model with the ZeroShot narrative reading function, we would need to generate a summary for all 10,784 passages in our tuning dataset. Since OpenAI’s pricing model charges per generated token, this was prohibitively expensive with the *davinci* model.

2. Entity Recognition

```
:: tell them to stay inside
 You go back out to the hallway, where your
     best friends are waiting.
"Are you crazy?" asks Robert. "You said it
    was a trap!"
"It probably is a trap," you admit. "But I
    can’t just leave them in there."
Maya just looks at you with big, sad eyes.
"Please," she says. "just this once."
You sigh. "I don’t know, Robert. I’m not
    going to be able to sleep if I know you
    two are in danger."
"Like you sleep anyway," says Robert. "What
    if something happens to us and you don’t
     even know?"
"Can’t we at least try to alert the [[unggoy
    ]] about the danger first?" you ask.

:: unggoy
 "Ain’t you the one who told me to ’keep my
     head up lowbrows’ and all that?"
[[Well, I’m just looking out for you.|
    another1]]
[[I have a solution.|another1]]
```

Figure 6: Passages generated via the entity recognition based reader from the same input passage as before.

For our next experiment, we define our narrative reading as

$$R(p) := (q_p, c_p, l_p) \tag{3}$$

where c was the set of characters referenced in the passage, l the set of locations, and q, the set of pronouns (“they”, “you”, “she”, etc.). These were extracted from passage text using the python *spacy* library for part of speech tagging and a pretrained BERT model[10] for state of the art entity recognition (Devlin et al. 2019). The explanation function X counted the characters and locations in the context chain and wrote the most frequent (top 8) of each to short sentences of the form: “Locations mentioned: ..., Characters mentioned: ... Pronouns referenced: ...”.

This model proved capable of generating passages that consistently referred back to previously mentioned locations and characters (Figure 6 mentions ‘Maya’ and ‘Robert’ from the previous passage), adding a sense of social and spatial

[10] https://huggingface.co/dslim/bert-base-NER

continuity to the stories. However, the passages generated still do not often follow the narrative, or if so, only loosely or by chance.

3. Event Extraction / Fabula

For the final experiment we report, we utilize the event extraction methodology of (Martin et al. 2018).

```
:: tell them to stay inside
 "It's dangerous out there!" you scream. You
     feel weak, but you manage to stand.
Maya and Robert look at you in shock. "You
    can't be serious!"
"I have to do something!" You say. "I have
    to protect everyone!"
You turn and rush back outside. You ignore
    the searing pain in your back and [[seek
     out the children]]

:: seek out the children
 You find a house with no roof and only
     walls left. What was once a house is
     now just a pile of rubble. You find a
     group of around 20 children huddled
     together. They are crying and shaking.
     What do you do?
[[you try to talk to them]]
[[you leave them alone]]
```

Figure 7: Passages generated via the fabula reader.

We expand on our previous narrative reader (Equation 3) with the following:

$$R(p) := (q_p, c_p, l_p, e_p) \qquad (4)$$

where e_p is an ordered list of (s, v, o) triples extracted from the passage.

X is defined similarly to the previous section, with the addition of a bulleted list of events, written in plain text as in Figure 8.

Thanks to modern NLP libraries, this reader architecture is cheap to run even on a CPU. Here, it does not directly predict event $e_{i+1}$from e_i, as in (Martin et al. 2018), who use a further LSTM model to expand predicted events into sentences. Rather, the pretrained language completion model, when fine-tuned on a corpus of $(prompt, response)$ pairs that include this event list, is expected to jointly learn next-event likelihood alongside Twine grammar.

To reduce the prompt length (necessitated by the discussion in the section *Formulating the Generation Problem*), we apply the following crude reduction algorithm to produce a new event list for X. We chose 32 as a constant following initial tests and intuition.

```
func reduce(events):
 while length(events) > 32:
   events = events[::2] // every other event
```

The effects of this necessary reduction step means that important events may be omitted from the fabula presented as context to the model. It is an active area of NLP research to find mechanisms to reduce the length of the prompt given to a text generation model while discarding only less important details. One may for example imagine a system for sifting events based on perceived relevance to the current passage.

```
<|begin|>Pronouns referenced: you, yourself,
     anyone, and everyone. Mentioned
    Locations: None. Mentioned People: Katie
    . Preceding Events:
* you packed bag
* you asked mom
* you tried not to look at anyone in the
    eyes
* you shove bag
* you use body weight
* you hear sound
* it gives way
* plane begins to move
* you feel hand<|title|>:: offering support
    <|start|>
```

Figure 8: An example prompt produced through the fabula reader.

However, the passages generated by this version of the model seem to appropriately introduce new characters and back-reference existing ones ('Maya', 'Robert' in Figure 6). It tends to stay on topic, with the deeper passage in Figure 6 appropriately following the tornado theme without overtly mentioning it. This example also demonstrates the model's tendency to follow the Twine convention of presenting branch points that represent straightforward character actions at the end of a passage, often weighted by considerations of exploration and morality.

Discussion

Generating Twine Stories

The project was originally motivated by asking if Twine *syntax* could be modeled in a straightforward way, and when GPT-3 was brought on board to bootstrap the process, it became clear that syntax was a trivial problem, inter-passage *fluency* was very high, and intra-passage *coherence* was the problem to solve. This required an exploration of the story generation literature. In iterating on our narrative reader formulation, we demonstrated that multiple theories of narrative can be substituted into this generative framework and that pairing the interpretive system with prompt-engineering is a fruitful methodology for language model based interactive story generation.

The Authoring Experience

Using the tool to write interactive fiction is itself a fun, game-like experience. The feeling of having primed the model to generate something interesting and coherent within the given narrative context is uniquely pleasurable, somewhat like having succeeded in having an interesting conversation with a stranger, resulting in the discovery of new opportunities for directions to take your writing. High coherence is not the only goal in building the model however; unexpectedness is also key (Table 1). The hope is that a tool like this might be useful for professional writers as well as novices looking for inspiration, writing assistance, and for those who would rather experience a custom Twine story than write one.

Passage Coherence	**User Response**
Incoherent; random	Annoyance, Confusion
Mildly Coherent; unexpected	Defamiliarization, Curiosity
Coherent; unexpected	Amusement, Joy, Ideation
Coherent; expected	Boredom

Table 1: Based on informal testing with 6 users, we identified a few common user responses to the various levels of coherence the models produce.

Limitations and Future Work

As we've established a method for using automated narrative readings to guide narrative text generation, it is clear that there are many additional theories of narrative that could be used instead of the fabula + entity method we arrived at. Such possible readings include formalist theories such as the SIG (Elson 2012), reader-response models (Castricato et al. 2021), or frame-based computational readings (Peng et al. 2021). Our earlier experimentation with other unsupervised summarization methods did not yield promising results, but a reviewer points out that this should be formally evaluated, and recent non-GPT abstractive summarization techniques such as those found in (Alomari et al. 2022) may suffice [11]). The fabula-based event structure we arrived at does not encapsulate a total understanding of narrative and we look forward to experimentation with many other representation formats.

Mawhorter et, al. has introduced the theory of *choice poetics*, "a formalist framework for understanding the impact of narrative choices on the player experience via their options, their outcomes, and how those relate to player goals" (Mawhorter et al. 2018). Its application to this work seems clear; we might model the sequence of choices that led to the present passage as an additional component of the reading.

Recent work has shown complex world state may be learned from a story via semantic role labeling, which can be used to populate knowledge graphs about the story world, and then stories can be generated such that they reduce differences between the current story world graph and a given goal graph (Peng et al. 2021). This approach is an extremely promising approach to story generation. Integrating a similar knowledge graph parsing approach into our architecture is an obvious next step, as is integrating a symbolic planner to reason over inferred narrative semantics.

Additionally, we are working towards a front-end interface that will allow for a more seamless revision-iteration loop for more lively or dynamic interaction with the written text. This will enable us to conduct a user study to assess the quality of the written work and the experience of working with the tool.

Finally, the evaluation of story text coherence beyond qualitative analysis needs to be addressed. Story coherence is generally assessed with human evaluation, though automated analysis of new character introduction or scene/object permanence may be possible. Without these evaluations, we are unable to make objective statements about the increase in narrative coherence we see from the baseline narrative reader to the fabula approach.

[11] such as RefSum (Liu, Dou, and Liu 2021) or gensim's TextRank summariser (`tinyurl.com/2s35hcr4`

Acknowledgements

We would like to acknowledge the Twine authors who made this work possible. In addition to creating the interactive narratives that were used for training our models, the community has been a wellspring of creativity and innovative storytelling for many years. We are very grateful to the Computational Media Department for nurturing interdisciplinary work like this and to the members of the Expressive Intelligence Studio, whose guidance and suggestions throughout the project were instrumental. We also thank our reviewers for their valuable feedback.

Author Contributions

AC wrote the paper, conducted research, and wrote the *Spindle* codebase. NWF guided the research and edited the paper. MM was the principle investigator, research advisor, edited the paper, and taught the graduate course on automatic game design (CMPM 244 at UC Santa Cruz) during which this project originated.

References

[Alomari et al. 2022] Alomari, A.; Idris, N.; Sabri, A. Q. M.; and Alsmadi, I. 2022. Deep reinforcement and transfer learning for abstractive text summarization: A review. *Computer Speech Language* 71:101276.

[Bal and Van Boheemen 2009] Bal, M., and Van Boheemen, C. 2009. *Narratology: Introduction to the theory of narrative*. University of Toronto Press.

[Bamman, Underwood, and Smith 2014] Bamman, D.; Underwood, T.; and Smith, N. A. 2014. A bayesian mixed effects model of literary character. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 370–379.

[Barros et al. 2019] Barros, G. A. B.; Green, M. C.; Liapis, A.; and Togelius, J. 2019. Who killed albert einstein? from open data to murder mystery games. *IEEE Transactions on Games* 11(1):79–89.

[Black and Wilensky 1979] Black, J. B., and Wilensky, R. 1979. An evaluation of story grammars. *Cognitive science* 3(3):213–229.

[Brown et al. 2020] Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

[Browne and Maire 2010] Browne, C., and Maire, F. 2010. Evolutionary game design. *IEEE Transactions on Computational Intelligence and AI in Games* 2(1):1–16.

[Calderwood et al. 2020] Calderwood, A.; Qiu, V.; Gero, K. I.; and Chilton, L. B. 2020. How novelists use generative language models: An exploratory user study. In *HAI-GEN+ user2agent@ IUI*.

[Castricato et al. 2021] Castricato, L.; Biderman, S.; Cardona-Rivera, R. E.; and Thue, D. 2021. Towards a formal model of narratives.

[Chambers and Jurafsky 2008] Chambers, N., and Jurafsky, D. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*, 789–797.

[Cho et al. 2018] Cho, W. S.; Zhang, P.; Zhang, Y.; Li, X.; Galley, M.; Brockett, C.; Wang, M.; and Gao, J. 2018. Towards coherent and cohesive long-form text generation. *arXiv preprint arXiv:1811.00511*.

[Compton, Kybartas, and Mateas 2015] Compton, K.; Kybartas, B.; and Mateas, M. 2015. Tracery: An author-focused generative text tool. In Schoenau-Fog, H.; Bruni, L. E.; Louchart, S.; and Baceviciute, S., eds., *Interactive Storytelling*, 154–161. Cham: Springer International Publishing.

[Cook, Colton, and Gow 2016] Cook, M.; Colton, S.; and Gow, J. 2016. The angelina videogame design system—part i. *IEEE Transactions on Computational Intelligence and AI in Games* 9(2):192–203.

[Cook 2015] Cook, M. 2015. Formalizing non-formalism: Breaking the rules of automated game design. *FDG*.

[Cook 2021] Cook, M. 2021. The road less travelled: Trying and failing to generate walking simulators. *arXiv preprint arXiv:2104.10789*.

[Côté et al. 2018] Côté, M.-A.; Kádár, A.; Yuan, X.; Kybartas, B.; Barnes, T.; Fine, E.; Moore, J.; Hausknecht, M.; Asri, L. E.; Adada, M.; et al. 2018. Textworld: A learning environment for text-based games. In *Workshop on Computer Games*, 41–75. Springer.

[Devlin et al. 2019] Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

[Duplantis et al. 2021] Duplantis, T.; Karth, I.; Kreminski, M.; Smith, A. M.; and Mateas, M. 2021. A genre-specific game description language for game boy rpgs. In *2021 IEEE Conference on Games (CoG)*, 1–8.

[Elson 2012] Elson, D. K. 2012. Detecting story analogies from annotations of time, action and agency. In *Proceedings of the LREC 2012 Workshop on Computational Models of Narrative, Istanbul, Turkey*, 91–99.

[Fan et al. 2020] Fan, A.; Urbanek, J.; Ringshia, P.; Dinan, E.; Qian, E.; Karamcheti, S.; Prabhumoye, S.; Kiela, D.; Rocktaschel, T.; Szlam, A.; et al. 2020. Generating interactive worlds with text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 1693–1700.

[Fan, Lewis, and Dauphin 2019] Fan, A.; Lewis, M.; and Dauphin, Y. 2019. Strategies for structuring story generation. *arXiv preprint arXiv:1902.01109*.

[Friedhoff 2013] Friedhoff, J. 2013. Untangling twine: A platform study. In *DiGRA conference*.

[Garbe et al. 2019] Garbe, J.; Kreminski, M.; Samuel, B.; Wardrip-Fruin, N.; and Mateas, M. 2019. Storyassembler: an engine for generating dynamic choice-driven narratives. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*, 1–10.

[Guan et al. 2021] Guan, J.; Mao, X.; Fan, C.; Liu, Z.; Ding, W.; and Huang, M. 2021. Long text generation by modeling sentence-level and discourse-level coherence. *arXiv preprint arXiv:2105.08963*.

[Harvey 2014] Harvey, A. 2014. Twine's revolution: Democratization, depoliticization, and the queering of game design. *G— A— M— E Games as Art, Media, Entertainment* 1(3).

[Howard and Ruder 2018] Howard, J., and Ruder, S. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

[Hua and Raley 2020] Hua, M., and Raley, R. 2020. Playing with unicorns: Ai dungeon and citizen nlp. *DHQ: Digital Humanities Quarterly* 14(4).

[Kostka et al. 2017a] Kostka, B.; Kwiecieli, J.; Kowalski, J.; and Rychlikowski, P. 2017a. Text-based adventures of the golovin ai agent. In *2017 IEEE Conference on Computational Intelligence and Games (CIG)*, 181–188.

[Kostka et al. 2017b] Kostka, B.; Kwiecieli, J.; Kowalski, J.; and Rychlikowski, P. 2017b. Text-based adventures of the golovin ai agent. In *2017 IEEE Conference on Computational Intelligence and Games (CIG)*, 181–188. IEEE.

[Lebowitz 1987] Lebowitz, M. 1987. Planning stories. In *Proceedings of the 9th annual conference of the cognitive science society*, 234–242.

[Lester, Al-Rfou, and Constant 2021] Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The power of scale for parameter-efficient prompt tuning.

[Liu and Chilton 2021] Liu, V., and Chilton, L. B. 2021. Design guidelines for prompt engineering text-to-image generative models.

[Liu, Dou, and Liu 2021] Liu, Y.; Dou, Z.-Y.; and Liu, P. 2021. Refsum: Refactoring neural summarization.

[Martin et al. 2018] Martin, L.; Ammanabrolu, P.; Wang, X.; Hancock, W.; Singh, S.; Harrison, B.; and Riedl, M. 2018. Event representations for automated story generation with deep neural nets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

[Mawhorter et al. 2018] Mawhorter, P.; Zegura, C.; Gray, A.; Jhala, A.; Mateas, M.; and Wardrip-Fruin, N. 2018. Choice poetics by example. *Arts* 7(3).

[Meehan 1977] Meehan, J. R. 1977. Tale-spin, an interactive program that writes stories. In *Ijcai*, volume 77, 9198.

[Pell 1992] Pell, B. 1992. Metagame: A new challenge for games and learning. *Heuristic programming in artificial intelligence*.

[Peng et al. 2021] Peng, X.; Xie, K.; Alabdulkarim, A.; Kayam, H.; Dani, S.; and Riedl, M. O. 2021. Guiding neural story generation with reader models.

[Pérez and Sharples 2001] Pérez, R., and Sharples, M. 2001. Mexica: A computer model of a cognitive account of creative writing. *Journal of Experimental & Theoretical Artificial Intelligence* 13:119 – 139.

[Pichotta and Mooney 2016] Pichotta, K., and Mooney, R. 2016. Learning statistical scripts with lstm recurrent neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

[Porteous 2016] Porteous, J. 2016. Planning technologies for interactive storytelling. In *Handbook of Digital Games and Entertainment Technologies*. Springer.

[Reynolds and McDonell 2021] Reynolds, L., and McDonell, K. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–7.

[Riedl and Young 2010] Riedl, M. O., and Young, R. M. 2010. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research* 39:217–268.

[Roemmele 2018] Roemmele, M. 2018. *Neural Networks for Narrative Continuation*. Ph.D. Dissertation, University of Southern California.

[Schaul 2013] Schaul, T. 2013. A video game description language for model-based or interactive learning. In *2013 IEEE Conference on Computational Inteligence in Games (CIG)*, 1–8. IEEE.

[Summerville et al. 2018] Summerville, A.; Martens, C.; Samuel, B.; Osborn, J.; Wardrip-Fruin, N.; and Mateas, M. 2018. Gemini: bidirectional generation and analysis of games via asp. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 14, 123–129.

[Theune et al. 2003] Theune, M.; Faas, S.; Nijholt, A.; and Heylen, D. 2003. The virtual storyteller: Story creation by intelligent agents. In *Proceedings of the Technologies for Interactive Digital Storytelling and Entertainment (TIDSE) Conference*, volume 204215, 116.

[Theune et al. 2004] Theune, M.; Rensen, S.; op den Akker, R.; Heylen, D.; and Nijholt, A. 2004. Emotional characters for automatic plot creation. In *International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, 95–100. Springer.

[Treanor et al. 2012] Treanor, M.; Blackford, B.; Mateas, M.; and Bogost, I. 2012. Game-o-matic: Generating videogames that represent ideas. In *Proceedings of the The Third Workshop on Procedural Content Generation in Games*, PCG'12, 1–8. New York, NY, USA: Association for Computing Machinery.

[Young et al. 2013] Young, R. M.; Ware, S. G.; Cassell, B. A.; and Robertson, J. 2013. Plans and planning in narrative generation: a review of plan-based approaches to the generation of story, discourse and interactivity in narratives. *Sprache und Datenverarbeitung, Special Issue on Formal and Computational Models of Narrative* 37(1-2):41–64.

Witscript 2: A System for Generating Improvised Jokes Without Wordplay

Joe Toplyn
Twenty Lane Media, LLC
P. O. Box 51
Rye, NY 10580 USA
joetoplyn@twentylanemedia.com

Abstract

A previous paper presented Witscript, a system for generating conversational jokes that rely on wordplay. This paper extends that work by presenting Witscript 2, which uses a large language model to generate conversational jokes that rely on common sense instead of wordplay. Like Witscript, Witscript 2 is based on joke-writing algorithms created by an expert comedy writer. Human evaluators judged Witscript 2's responses to input sentences to be jokes 46% of the time, compared to 70% of the time for human-written responses. This is evidence that Witscript 2 represents another step toward giving a chatbot a humanlike sense of humor.

Introduction

To be truly enjoyable and credible, a conversational agent like a social robot needs to produce contextually integrated jokes about what's happening at the moment (Ritschel, Aslan, Sedlbauer, and André 2019).

Existing computational humor systems can generate conversational jokes that depend on wordplay. But generating jokes that don't rely on wordplay has proven to be more difficult. Indeed, generating all types of humor is often regarded as an AI-complete problem (Winters 2021). Nevertheless, Witscript 2 is a novel system for automatically generating contextually integrated jokes that are based not on wordplay, but on common sense.

Related Work

Few systems for the computational generation of verbally expressed humor can generate contextually integrated jokes, such as jokes improvised in a conversation. The term verbally expressed humor is used here to mean humor conveyed in language, as opposed to verbal humor, which is sometimes used to mean humor that depends on wordplay (Ritchie 2000).

The method of Zhang, Liu, Lv, and Luo (2020) generates a punch line given a set-up sentence and relevant world knowledge. But its effectiveness is limited because it does not incorporate explicit humor algorithms. The system of Ritschel et al. (2019) transforms a non-ironic utterance into a humorously ironic version using natural language processing and natural language generation techniques. Zhu (2018) uses search engine query statistics to generate a response to a user's utterance that is humorously improbable given the subject of the utterance. The PUNDA Simple system (Dybala, Ptaszynski, Higuchi, Rzepka, and Araki 2008) and the Witscript system (Toplyn 2021) generate joke responses in a conversation, but those jokes rely on wordplay.

In contrast, the Witscript 2 system uses explicit humor algorithms to generate, in real time, conversational jokes that rely on common sense instead of wordplay.

One of those humor algorithms, derived from the Surprise Theory of Laughter (Toplyn 2021), specifies that a monologue-type joke has these three parts:

1. The **topic** is the statement that the joke is based on.
2. The **angle** is a word sequence that smoothly bridges the gap between the topic and the punch line.
3. The **punch line** is the word or phrase that results in a laugh. It's an incongruity at the end of the joke that, surprisingly, turns out to be related to elements of the topic.

Witscript 2, like Witscript, incorporates the Basic Joke-Writing Algorithm (Toplyn 2021), which consists of five steps for writing a three-part joke:

1. **Select a topic.** A good joke topic is one sentence that is likely to capture the attention of the audience.
2. **Select two topic handles.** The topic handles are the two words or phrases in the topic that are the most attention-getting.
3. **Generate associations of the two topic handles.** An association is something that the audience is likely to think of when they think about a particular subject.
4. **Create a punch line.** The punch line links an association of one topic handle to an association of the other topic handle in a surprising way.
5. **Generate an angle between the topic and punch line.** The angle is text that connects the topic to the punch line in a natural-sounding way.

Now I'll describe how the Witscript 2 system executes the five steps of the Basic Joke-Writing Algorithm.

Description of the Witscript 2 System

The Witscript 2 system is powered by a large language model, OpenAI's GPT-3 (Brown et al. 2020). The most capable GPT-3 model currently available is used, the text-davinci-002 model. GPT-3 was trained on a filtered version of Common Crawl, English-language Wikipedia, and other high-quality datasets. I accessed GPT-3 via the

OpenAI API (`https://openai.com/api/`) and did not fine-tune the model.

Here's how Witscript 2, using GPT-3 components, executes the Basic Joke-Writing Algorithm to generate a joke response to a conversational topic:

1. **Select a topic.** Witscript 2 receives a sentence from a user and treats it as the topic of a three-part joke. For example, the user tells the system, "The U.S. is planning to buy 22 aging fighter jets from Switzerland."
2. **Select two topic handles.** The GPT-3 API is called with a prompt to select the two most attention-getting nouns, noun phrases, or named entities in the topic. From that example topic, GPT-3 selects the topic handles "fighter jets" and "Switzerland."
3. **Generate associations of the two topic handles.** The GPT-3 API is called with a prompt to generate a list of associations for each topic handle. In our example, for "fighter jets" GPT-3 generates a list including "F-22 Raptor." For "Switzerland" it generates a list including "Swiss chocolate."
4. **Create a punch line.** The GPT-3 API is called with a prompt to select one association from each list and combine them. In our example, GPT-3 selects "F-22 Raptor" and "Swiss chocolate" and combines them to create the punch line "Swiss Chocolate F-22s."
5. **Generate an angle between the topic and punch line.** The GPT-3 API is called with a prompt to generate a joke, based on the topic, that ends with the punch line. In our example, the system generates the joke "I hear they're delicious Swiss Chocolate F-22s." The system responds to the user with the joke.

System Evaluation

To evaluate Witscript 2, I used the same methodology and 13 input topics that were used to evaluate Witscript (Toplyn 2021). For a new baseline, I used the text-davinci-002 model of GPT-3; for each input topic, the GPT-3 API was called with the prompt "Generate a short response."

I hired workers via Amazon Mechanical Turk (AMT) to evaluate the responses generated by GPT-3 and Witscript 2, and also the responses of Witscript and a human comedy writer that are listed by Toplyn (2021). None of the 13 input topics were used in any prompts given to the GPT-3 API. I specified that the AMT workers had to be located in the United States and have a Human Intelligence Task (HIT) Approval Rate greater than or equal to 95%.

The 52 input and response pairs were put in a random order and presented in that same order to all of the AMT evaluators, who were not told how the pairs had been obtained. Given each input and response pair, the evaluators were asked to rate the response as either 1 (not a joke), 2 (almost a joke), 3 (a joke), or 4 (a very good joke).

Each input and response pair was evaluated by 15 AMT workers (780 ratings total). Table 1 shows the first response that was output by the three systems for each input, along with the human's response. All the responses were transcribed exactly as they were output. The mean rating given by the evaluators for each response is also shown.

Source	Input and responses	Rating
	1. Input: A man pulled out a gun at a Popeyes restaurant when he heard they had no more chicken sandwiches.	
GPT-3	This is ridiculous! People need to learn to control themselves.	1.80
Witscript	Popeyes gun? Um, not a Bubba Gun.	2.53
Witscript 2	The man was arrested and charged with "attempted Cajun Fried Chicken."	2.47
Human	He also objected to any new laws on bun control.	2.40
	2. Input: America’s national forests are being severely damaged by the illegal growing of marijuana.	
GPT-3	This is a huge problem that needs to be addressed.	1.53
Witscript	America marijuana? Yah, so now he wants to be not in Canada cannabis.	2.20
Witscript 2	The worst part is the people doing it are just "smoking weed in the trees."	1.73
Human	Which explains the new Forest Service mascot, Tokey the Bear.	2.33
	3. Input: For Oktoberfest, Adidas is releasing sneakers that repel beer and vomit.	
GPT-3	That's pretty slick!	2.27
Witscript	Adidas vomit? Ah, I like the Puma poop.	2.80
Witscript 2	Budweiser Puke.	2.07
Human	And they’re changing their slogan to “Just Spew It.”	2.60
	4. Input: Two men were arrested in the theft of a hundred thousand diapers.	
GPT-3	This is a serious crime. If convicted, the two men could face significant jail time.	1.60
Witscript	Theft diapers? Like, they stole stroller.	2.00
Witscript 2	The headline should have read: "Pampers & Robbers."	2.80
Human	They were charged with potty larceny.	2.93

Table 1: Responses generated for each input, with the mean rating of each response on a scale from 1 (not a joke) to 4 (a very good joke).

	5. Input: It's the twenty-fifth anniversary of the Blue Man Group.	
GPT-3	That's so cool!	1.60
Witscript	Blue anniversary? Heh, welcome to the Bluebilee.	2.73
Witscript 2	And they're celebrating their Blue Wedding Anniversary.	2.13
Human	To celebrate, they’re starting their own health plan: Blue Man Blue Cross Blue Shield.	2.87
	6. Input: Thirteen TGI Fridays were selling cheap liquor in premium brand bottles.	
GPT-3	That's a lot of TGI Fridays.	1.93
Witscript	Thirteen Fridays? Okay, but I prefer a nine night myself.	2.67
Witscript 2	The good news is that they're now serving "Drunken Wings."	2.47
Human	The company claims it’s a new promotion: TGI Fraudays.	2.73
	7. Input: At a zoo, the first rhinoceros ever conceived by artificial insemination was born.	
GPT-3	How wonderful!	1.53
Witscript	Rhinoceros insemination? Mmm-hmm, welcome to the zoosemination.	1.93
Witscript 2	The baby rhino's parents are so excited, they can't wait to show him off to their friends. They're going to take him to the in vitro rhinoceros horn.	2.27
Human	Usually rhinos are conceived using that dating site for large mammals, Ashley Mastodon.	2.53
	8. Input: Dow Chemical is turning raw sewage into drinking water.	
GPT-3	Dow Chemical is turning raw sewage into drinking water. This is a great way to recycle water and help save the environment.	1.87
Witscript	Dow Chemical drinking? Yup, so not Dowing.	2.33
Witscript 2	I'm not sure if that's a good thing or a bad thing, but they're calling it the 'Dow Jones Industrial Fountain.'	1.80
Human	To make it more appealing, they’re calling it Pooland Spring.	3.13
	9. Input: A new study says drinking sugar-sweetened beverages increases the risk of death.	
GPT-3	This is yet another reason to avoid sugary drinks!	1.73
Witscript	Beverages risk? Huh, I have to focus on drinks risks.	2.20
Witscript 2	The study was sponsored by "Coca-Cola funeral."	2.07
Human	That's why they're renaming that soft drink Coca-Coma.	2.53
	10. Input: In one of its restrooms, the Guggenheim Museum is installing a solid gold toilet.	
GPT-3	This is an interesting bit of news.	1.80
Witscript	Gold toilet? Yep, to make a toilet pure.	2.33
Witscript 2	"The Golden Throne." Yeah, it's a little gaudy, but it's perfect for a museum that's already full of crap.	2.93
Human	It’s perfect for people who have just eaten 24 carrots.	3.20
	11. Input: A brewery apologized for a beer that shares the same name as a Hindu deity.	
GPT-3	The brewery has issued an apology and is taking steps to rename the beer.	1.73
Witscript	Shares Hindu? Heh, no Sikh stock.	2.07
Witscript 2	I guess they should have seen that one coming. Now they're getting hammered for "Beer Brahma."	2.33
Human	No word yet on their plans for their other beer, Buddha-weiser.	2.73
	12. Input: A truck was stolen that contained half a million dollars' worth of handbags.	
GPT-3	This is a serious crime with significant financial implications. The truck and its contents should be recovered as soon as possible.	1.73
Witscript	Truck handbags? Yep, of vanbags.	2.13
Witscript 2	The truck was a Ford Vuitton.	2.73
Human	The thief was arrested and the handbags were charged as accessories.	3.07
	13. Input: A man was caught during a home burglary wearing only lingerie.	
GPT-3	This is an outrageous story!	1.67
Witscript	Burglary lingerie? Yes, and the burglaries bikinis.	2.47
Witscript 2	Victoria's Secret police report.	2.60
Human	He was charged with panty larceny.	2.93

Table 1 (continued).

Table 2 compares the systems based on their ratings. The second column shows that Witscript 2's responses were rated, on average, about halfway between those of the GPT-3 baseline and the human, a professional comedy writer. Witscript 2's responses were also rated, on average, the same as Witscript's responses, a result that may not seem particularly impressive. But that result is encouraging because it shows that Witscript 2 can create jokes that are as successful as, but more sophisticated than, mere wordplay jokes, which are usually regarded as the low-hanging fruit of computational humor.

The last column of Table 2 shows the percentage of responses that the evaluators rated as "a joke" or "a very good joke." Witscript 2's responses were judged to be jokes 46% of the time, compared to only 25% of the time for the GPT-3 baseline responses. This result, too, is encouraging because it is additional evidence that a system to generate contextually integrated jokes is feasible.

System	Mean rating	% jokes (ratings of 3 or 4)
GPT-3	1.75	25.1%
Witscript	2.34	47.2%
Witscript 2	2.34	46.2%
Human	2.77	70.3%

Table 2: Comparison of the systems based on their ratings.

Discussion

Computational Creativity

I believe that the Witscript 2 system demonstrates computational creativity instead of mere generation because its output exhibits three characteristics: novelty, value, and intentionality (Ventura 2016).

The system's output has **novelty** because each contextually relevant joke that the system improvises in response to a new input has almost certainly never been created before by it or by any other agent.

The system's output has **value** in that human evaluators judge the system's responses to be jokes 46% of the time, and conversational jokes like those output by the system have worth and usefulness (Dybala et al. 2008).

And the system produces that novel, valuable output with **intentionality** in several ways: It restricts its generation process by using domain knowledge about how a professionally-written joke is structured. It generates jokes in an autonomous fashion by using a language model prompted with an inspiring set consisting of quality examples, namely professionally-written jokes. Finally, it apparently employs a fitness function to intentionally filter out joke responses that don't meet some threshold of value.

For example, given the topic "Today the Arby's fast food chain announced the release of a vodka that tastes like their French fries," Witscript 2 responded, "The good news is, now you can get drunk and fat at the same time." In doing so, it deliberately rejected the punch line that it had generated using the Basic Joke-Writing Algorithm: "Smirnoff and McDonald's." Instead, it improvised a different punch line and created a joke that it somehow decided was more worthy of being output.

Commonsense Knowledge

In addition to computational creativity, Witscript 2 demonstrates commonsense knowledge. This commonsense knowledge consists of knowledge of everyday commonsense relations.

For example, in generating the joke about the fighter jets from Switzerland, Witscript 2 exhibits taxonomic reasoning (Davis and Marcus 2015) when it infers that "F-22 Raptor" is an instance of "fighter jets." In terms of the commonsense relation types in the commonsense knowledge graph ATOMIC 2020 (Hwang et al. 2021), Witscript 2 humorously infers that a physical object from Switzerland would be made of ("MadeUpOf ") a material commonly found in ("AtLocation") Switzerland, i.e., Swiss chocolate. Witscript 2 also infers that a physical object made of Swiss chocolate would be ("HasProperty") delicious.

Contributions

This paper makes the following contributions:

1. It introduces a novel system for automatically improvising contextually integrated jokes that don't depend on wordplay.
2. It shows how computational humor can be implemented with a hybrid of a large language model and symbolic AI, where the symbolic AI incorporates expert knowledge of comedy domain rules and algorithms.
3. It demonstrates that generating humor that relies on some commonsense knowledge may not be an AI-complete problem.

Future Work

I anticipate that future work will improve the performance of Witscript 2 until its jokes based on common sense are rated better than the wordplay jokes of Witscript. To that end, work will be directed toward getting Witscript 2 to execute the Basic Joke-Writing Algorithm more effectively.

To accomplish that, the following will be explored: using different prompts and configuration settings for base GPT-3 models; fine-tuning base GPT-3 models to create multiple customized versions, each version optimized to carry out one joke-writing step; and substituting different large language models for GPT-3.

Conclusion

The Witscript 2 joke generation system could be integrated into a chatbot as a humor module; the proprietary software is available for license. Such a humor-enabled chatbot might potentially animate an artificial, but likeable, companion for lonely humans.

Acknowledgments

The author would like to thank three anonymous reviewers for their helpful comments.

Author Contributions

J. T. ideated and wrote the paper alone.

References

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T. J.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. ArXiv, abs/2005.14165.

Davis, E., and Marcus, G. 2015. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM.* Vol. 58, No. 9. 92–103.

Dybala, P.; Ptaszynski, M.; Higuchi, S.; Rzepka, R.; and Araki, K. 2008. Humor Prevails! - Implementing a Joke Generator into a Conversational System. In Wobcke, W.; and Zhang, M., eds., *AI 2008: Advances in Artificial Intelligence (AI 2008)*. Lecture Notes in Computer Science, vol. 5360. Berlin: Springer.

Hwang, J. D.; Bhagavatula, C.; Le Bras, R.; Da, J.; Sakaguchi, K.; Bosselut, A.; and Choi, Y. 2021. COMET-ATOMIC 2020: On Symbolic and Neural Commonsense Knowledge Graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(7), 6384–6392.

Ritchie, G. D. 2000. Describing Verbally Expressed Humour. *Proceedings of AISB Symposium on Creative and Cultural Aspects and Applications of AI and Cognitive Science*, 71-78. Birmingham, UK.

Ritschel, H.; Aslan, I.; Sedlbauer, D.; and André, E. 2019. Irony Man: Augmenting a Social Robot with the Ability to Use Irony in Multimodal Communication with Humans. *AAMAS*.

Toplyn, J. 2021. Witscript: A System for Generating Improvised Jokes in a Conversation. In *Proceedings of the 12th International Conference on Computational Creativity*, 22–31.

Ventura, D. 2016. Mere Generation: Essential Barometer or Dated Concept? In *Proceedings of the Seventh International Conference on Computational Creativity*, 17–24. Sony CSL, Paris.

Winters, T. 2021. Computers Learning Humor Is No Joke. *Harvard Data Science Review*, 3(2).

Zhang, H.; Liu, D.; Lv, J.; and Luo, C. 2020. Let's be Humorous: Knowledge Enhanced Humor Generation. *ACL*.

Zhu, D. 2018. Humor robot and humor generation method based on big data search through IOT. *Cluster Computing*, 1-7.

Two-Fisted Comics Generation:

Comics as a Medium and as a Representation for Creative Meanings

Tony Veale,

School of Computer Science, University College Dublin, Ireland.

Tony.Veale@UCD.ie

Abstract

Every expressive medium allows us to ground meaning in different ways. Comics, or the so-called 9^{th} *art* (after film and TV), are sequential integrations of words and images that offer more possibilities than either words or images alone can offer. Like any art form, comics settle into clichéd norms – panels, balloons, tails – and give rise to genres that dominate at the expense of all others, such as the superhero genre. Yet comics can also use a vivid emotional expressivity to imbue physical actions with potent feelings, lending intuitive immediacy to the gamut of human concerns. This paper considers comics as both a medium for automated story-telling and as a meaning representation from which machines can shape specific meanings for particular audiences. We propose two XML standards for communicating via comics: one to define an underlying narrative, and another to define the comics derived from it. The latter uses a repository of visual assets to convey actions and emotions, placing posable characters against a backdrop of stock settings. We show how a combinatorial approach accommodates all of the outputs of an automated story-generator, and also explore how it adapts to the very human exchanges of an online debate, such as the fractious Twitter debate on vaccines. Comics appear to be a practical medium in which to make targeted interventions into a polarizing debate, to present opposing points of view to each side.

See You In The Funny Pages

Although frequently packaged in a disposable form, comics have been described as a *sequential art* by Eisner (1985) – for whom the Eisner prize in comics is named – and as *the ninth art* by Maurice De Bevere, the creator of *Lucky Luke*. At its simplest, a comic strip is a sequence of framed snap shots, called panels, separated by thin bands of whitespace, called gutters. Each panel is usually square or rectangular, typically framed in a black border, and sometimes labeled with a caption above or below the scene depicted within. A typical panel contains a mix of textual and visual elements, to depict a specific action in a certain setting, and to record any words that are spoken (or privately thought) in context. Those words are most often contained within text balloons, either rounded speech balloons or fluffy, cloud-like thought balloons, whose tails link them to the vocalizing characters.

These conventions have become the stuff of cliché, but as McCloud (1993) has shown, this normative grammar of comics allows for a great deal of divergence and creativity. Indeed, even text balloons can vary tremendously from one context to another, to shape meaning as well as just contain it (Forceville *et al*., 2010). Although the ease with which children adapt to the medium's mechanisms allows some to dismiss it as juvenile, this ease also reflects the depth of the cognitive foundations in which the medium is rooted (Cohn, 2013; Cohn & Magliano, 2020). For instance, one intuitively reads a comic strip in the order one reads a text, from top to bottom and left to right in the West, and from right to left and back to front in the East. No one needs to be taught how to read a comic strip. Readers simply adapt to the blended medium as a new form as visual reading.

This work explores the automated generation of comic strips as containers and communicators of meaning. While the marriage of visual and textual forms makes comics an ideal medium for computational creativity, our aim is to do more than produce comic-shaped outputs that readers may find attractive in their own right. Rather, our aim is to use the comic strip as a means of communication in which we pour meaning from one kind of container – such as a text – into another – a comic combining images *and* words – with no, or little, loss of meaning. Our goal is not an end-to-end production of comics in which interior levels of meaning go unexposed to scrutiny and manipulation by the producer, but a controlled, meaning-preserving translation from one explicit representation of meaning to another. To this end, we present a comics-generator that works with the outputs of an automated story-generator, translating each tale from a purely textual form to a vivid blend of words and images.

Although comics are entertaining in their own right, we explore a practical use of the medium here. Consider why they are called 'comics' or 'funny-books' in the first place: the name is a carry-over from the earliest newspaper strips in which short, whimsical diversions were illustrated (the first American "comic book", *Famous Funnies*, repackaged these newspaper funny pages as a standalone periodical). Even serious comicbooks – which some now call *Graphic Novels* – still carry traces of the comical and the unserious. The larger context of this work makes use of these vestiges to package polarizing and perhaps unwelcome meanings in more welcome and disarming forms. Those meanings arise in heated online debates, such as the Twitter debate around vaccines, in which disputants on each side show a tendency to dig in, tune out and listen only to those on the same side. Machines can help to break down these echo chambers by making targeted interventions into the debate, using comics to summarize and distill the main arguments on both sides. As a first step, we will examine how well a framework for producing comics from computer-generated tales can also produce comics from these arguments, preserving the gist of each argument and the gist of any given user's position.

With these goals in mind, the rest of the paper assumes the following structure. After exploring some related work and ideas in the next section, we present our combinatorial approach to comics production, which maps from an XML schema for machine stories to an XML schema for comics. We next consider applications of this mapping of XMLs, in dialogue-driven comics production and online intervention. For the latter, comics must be attuned to the dynamics of a debate as reflected in a representative dataset, so we model the debate via statistical analysis of a large Twitter corpus. Our analysis of the vaccine debate will show that a comics creator that is attuned to a sufficiently rich story creator is capable, through the liberal use of visual metaphor, to also accommodate the diverse arguments of a topical debate.

Related Work and Ideas

Comics are a sequential art that requires a narrative impetus. A generator can produce this impetus for itself, by creating its own stories, or it can acquire its narratives from another source, such as an existing story-generator, or from another medium, such as film (by e.g. reusing film scripts), theatre, online discussions (e.g., chatrooms, Twitter), or games. For example, a comic narrative might visualize the sequence of moves in a chess game as a sequence of real-world actions (Gervás, 2014), or mirror the sequence of moves in a video game. In the latter, game screenshots might also be used to provide the visual contents of the comic's panels.

The *Comic Chat* system of Kurlander *et al.* (1996) takes its narrative impetus from chatroom interactions, and turns those textual discussions into comic strips, filling one panel per conversational turn. Each interacting user is assigned a stock comic figure, such as a lantern-jawed jock, an exotic princess, or an anthropomorphic animal, where each figure has a small number of facial expressions and neutral poses. Those expressions, used to convey the basic emotions, are determined via a sentiment analysis of a user's contribution to a turn, which floats overhead in a text balloon. Because this narrative impetus tracks the human inputs, *Comic Chat* is free to focus on the technical craft of the medium, and it shows a firm grasp of the grammar of comics. It uses long-shots to start a conversation, and close-ups for subsequent turns. All figures, balloons and tails are intuitively ordered within a panel to ensure ease of reading from left to right, and a small number of backgrounds is used consistently to maintain continuity from one panel to the next.

The *Comics2D* system of Alves *et al.* (2007) builds its comics from the storyworld representations of a generator of dramatic fiction, such as that of Cavazza *et al.* (2003). If a fiction generator does more than generate narative texts, and also provides a visual representation of its story-world, a comics generator can tap into this visual model too, to fill its panels with snapshots of the story. Comics2D uses its own XML representation of a comic, via a schema it calls CSDL (or *Comic Strip Description Language*). The schema defines nodes for each of the principal elements of a comic, from panels and scenes to backgrounds and characters, and also explicitly tags what happens in the transitions between panels. *Comics2D* is a modular system that allows users to plug-in alternate renderers, and it should support any story-generator than works with CSDL, although the relationship between renderer and generator is typically a complex one.

A comic strip is a sequence of snapshots held together by the driving logic of a story, but this logic often lies hidden in the gutters between panels. Data-rich machine learning approaches can teach a machine to infer this logic, so that it can predict for itself what should come next in the story. To this end, Iyyer *et al.* (2017) have created their COMICS dataset from 1.2M comic panels, for which the text within is automatically transcribed. They estimate that most panel transitions are either *action-to-action* (~34%) or *subject-to-subject* (~32%), while ~17% extend a conversation, ~14% shift from one scene to another, and less than 1% illustrate the moment-to-moment dynamics of a single action. Iyyer *et al.* train a hierarchical network of LSTMs to derive a context model for a given sequence of panels, and use this model to score candidates for the words and images in the subsequent panels. Although the model still underperforms humans, it showcases the value of a multi-panel context and a multimodal integration of visual *and* textual features.

Such a model might, in principle, also generate the next panel, and not just prefer one or another panel continuation. Melistas *et al.* (2021) use *two* neural architectures to create comics in the style of Japanese manga: a language model, *GPT-2*, to produce the text of each panel, and a generative adversarial network, *StyleGAN2*, to synthesize the images. As with Iyyer *et al.*, they create a dataset of manga comics for which textual transcriptions are automatically extracted. Text areas are then inpainted to yield text-free training data for the image synthesizer, whose training is further boosted with images of Japanese anime. Low-resolution outputs are then subsequently refined with a trained upscaling network. The approach is suggestive and highly experimental, as the generative models for text and image operate independently of each other, to produce sequences of images that have no pre-conceived relation to the text balloons that adorn them.

Agrawal *et al.* (2016) show it is feasible to make comics from a jumble of captioned photos, by using a mix of word and image features to infer the most natural narrative order. Those photos can, in turn, serve as a basis for generating a comics-style rendering for each photo/panel. Yet, as useful as these rich datasets are for machine-learning approaches, they lack a key dimension: a sense of the underlying story that gives the images and text their narrative momentum.

A Combinatorial Approach

Melistas *et al*. use a neural blackbox to generate the textual content of a comic strip in a single pass, so its output is a surface form that lacks an overt deep structure. A generator that produces XML-tagged surface forms can be subjected to tests of well-formedness and schema-specific validity, so that ill-formed or invalid outputs can simply be resampled, but raw surface forms can offer no such affordances. Multi-pass approaches that first generate a deep 'fabula' structure before producing the surface rendering of the story – as text for a narrator and dialogue for the characters – offer even more control to the creator of comic strips. This fabula can be mapped to the panel layout of the comic, while narration text can provide the panels' captions, and dialogue text can provide the contents of the panels' balloons. What may now seem like old-fashioned approaches to story-generation are thus well-placed to support step-by-step comics production.

Any story-generator that provides an explicit fabula built from a fixed inventory of action types, a surface rendering of the story, and dialogue for its characters, is well suited to automatic comic generation. While there are many story-generators that potentially fit this bill, from that of Cavazza *et al*. (2003) to Montfort *et al*. (2013) and Gervás (2014), we make use of the *Scéalextric* system here (Veale, 2017). *Scéalextric* structures each story as a sequence of "beats." As defined by Gaiman (2021), a *beat* is the smallest unit of action in a story or comic, a discretely resolvable event that is worthy of textual or visual rendering. A beat focalizes a moment in time, and is the ideal unit of comics structure. Each *Scéalextric* beat comprises a single two-person action from its inventory of 800 distinct types, which encompass the realms of romance, crime, medicine, politics, business, religion and war. Each beat is rendered as a 3rd-person view of the event, which can serve to caption the corresponding panel, and provides spoken (or internal) dialogue for each of the two participants, which can fill the panel's balloons.

We define a scene as a sequence of beats that focalize the same characters. A scene may juggle several characters, but each beat will focus on just two at a time. As the characters move in and out of a setting, the scene changes. A dramatic change typically tracks a change in location, and a change in background for the corresponding comic panels. Such a change may warrant a visual flourish such as a splash panel or an establishing shot for the new location, while a lower key shift may warrant just a close-up on a single character.

To tease apart the varying concerns of story and comic, we define two XML formats, one for each. *ScéalXML* is the schema that captures the nested layers of a *Scéalextric* tale, embedding the textual substance of narration and dialogue within a beat structure that also defines fabula-level events. This schema can, in principle, be used to encode the stories of other generators, either *as is* or with some extensions, so that the same mapping to *ComiXML*, and then onwards to a rendered comic strip, can serve those other generators also. *ComiXML* is the schema that encodes a comics-level view of the same events. *ScéalXML* composes scenes from beats, while *ComiXML* composes analogous chapters from panels. Although designed to mirror *ScéalXML* in the first instance, *ComiXML* has obvious overlaps with the CSDL schema of Alves *et al.* (2007), and with the CBML (or *Comic Book Markup Language*) of McIntosh (2005) and Walsh (2012).

In addition to parallel scene/chapter and beat/panel nodes, *ScéalXML* & *ComiXML* also define character/figure nodes. In a story, a <*character*> element names a recurring entity that participates in one or more beats as either the *agent* or the *patient* of an action. Nodes of this type also specify long and short names for a character, as well as the pronouns for referencing them obliquely. *Scéalextric*'s cast of thousands comprises well-established personae from fact and fiction, such as Cleopatra, Darth Vader, Bill Gates and Maleficent. This adds colour, and a potential for humorous incongruity, to the textual outputs of *Scéalextric*, but it poses a challenge for any comics application that must render them visually.

This challenge is two-fold: we need to render characters in ways that are recognizable, or at least differentiable, and we need to render them in emotionally expressive poses that reflect our intuitions of how one performs a specific action. We also need to imbue the outputs with an inherently zany or whimsical charm; a comic may have a serious intent, but to disarm a skeptical audience it must also appear flippant. We meet this challenge with a bespoke set of visual assets called *funny-bones*. Unlike those of Kurlander *et al.* (1996), which attach emotive heads to neutral bodies, these assets integrate a strong emotion with a vivid pose, since a feeling is not separable from the actions that cause it. Each funny-bone has a large expressive head with short rubber limbs, and modifiable hair, skin and lips that allow each to depict a male or female *Scéalextric* persona. For instance, Cleopatra is pre-defined with long black hair, olive skin and red lips, Julius Caesar is given short white hair, pale skin and pink lips, and a bald or nearly bald character, such as Joe Biden, is simply given short, skin-toned hair. We do not shoot for accuracy, just a cartoonish sense of who these people are.

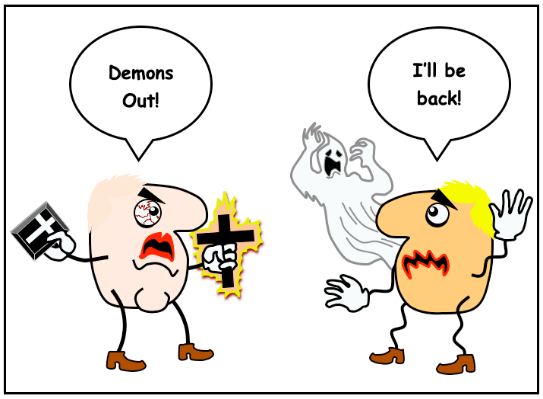


Fig. 1. A single panel with two funnybone figures, configured as *Joe Biden* and *Donald Trump*, in poses *exorcising* and *exorcised*.

Fig. 1. presents a panel for the single *Scéalextric* story beat, <*Joe_Biden exorcises Donald_Trump*>. Its two figures face each other against a blank setting, and are configured with appropriate hair, skin and lip values. In the XML encoding of this <*panel*> node, each is also specified with a location (*left* or *right* of panel), orientation (facing *left* or *right*), and a balloon node (*speech* or *thought*) with an apt text filling.

We consider the emotions expected of the agent and patient roles for each of the 800 *Scéalextric* actions, and how best to render them in mid-action. This leads us to produce 230 funny-bone assets – e.g., an angry figure attacking, a scared figure running away, a strident figure with a puffed chest – and assign one or more to each role of all 800 actions. These 230 assets serve, for the most part, as visual metaphors that concretize and exaggerate the action. So, destructive figures brandish giant hammers or axes; doctors wield scalpels and syringes, and mad scientists cackle as rats scurry underfoot. Zealous figures thump bibles or hurl fireballs, and sick ones slump in wheelchairs as sad ones stand under storm clouds. The assets favour immediacy over nuance, drama over tact. The same is true of the 100 backdrop images that we create to anchor an actor to the domain of their action. Backdrops include hospitals (interior and exterior), dungeons and labs, churches and offices, outdoor scenes (parks, farms, streets), edifices (police stations, court houses, banks, government buildings, jails) as well as battlefields, cafés, gyms and bars. These are mapped both to *Scéalextric* actions and to specific figure assets. The first mapping allows the system to pick a backdrop for a pre-defined action, and the second allows it to infer a suitable choice for any arbitrary pairing of figures. For instance, a stalker lurking in the bushes (*stalking* pose) must be placed against an outdoor scene, not an indoor one.

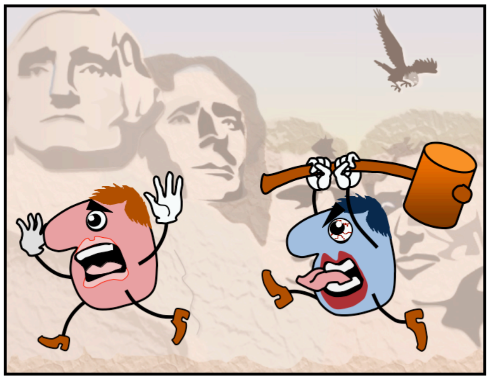


Fig. 2. A panel with "*Mount Rushmore*" as its backdrop. The two figures are colour-coded to visually represent *red* and *blue* states.

The <*setting*> element is used to add a background asset to a panel in ComiXML. As illustrated in Fig. 2, which shows one panel of a comic for the hashtag *#DemsAreDestroying America*, backdrops add another degree of metaphoricity to a panel's meaning. Here the background is a character in its own right, with the *Mt. Rushmore* setting used to depict the USA; the two figures are further colour-coded to represent *blue states* (i.e. "Dems") and *red states* (i.e. "not Dems"). A backdrop can be used literally, to denote a real setting such as a bar or kitchen, or metaphorically, as when a *battlefield* scene evokes bitter enmity, or a *police state* (barbed wire and searchlights at night) is used to denote oppressiveness.

Applications

These assets are sufficient to give every *Scéalextric* story a comic-strip form. The general application, named *Excelsior*, can sample *Scéalextric*'s ouevre at random, or retrieve tales that involve a given character, such as *Oprah* or *Bill Gates*. Examples of *Excelsior*'s comics for two *Scéalextric* stories are accessible online.[1] Its comics can be rendered as HTML documents that are suited to direct publication on the web, or as animated GIFs that may be shared within a tweet.

A co-creative variant of *Excelsior* engages in a dialogue with the user to refine the comic that is jointly produced. A metaphor-oriented chatbot, named *Figaro*, is repurposed to manage this dialogue using its rich lexicon and a system of pattern-matching rules. The narrative impetus for the joint work is at first provided by the user, in the form of a simple statement, typically a metaphor, that establishes the action, such as "my life is a joke" or "Donald is in the dog house." As in most chatbots, this input is then mapped via a series of stimulus:response rules into a corresponding output text. However, *Excelsior* does not generate raw surface outputs but XML forms that integrate a story-level understanding of the input with a ComiXML rendering of that interpretation.

It does this by tapping into the *Scéalextric* causal graph, which links every story action to every possible next action via *so*, *but* and *then* arcs. The Figaro lexicon connects the words of the user's input to relevant vertices in this graph, so that e.g., "life" maps onto *interact_with* (agent), "joke" maps to *is_entertained_by* (patient) and *laugh_at* (patient), and "in the dog house" maps to *criticize* (patient), *chastise* (patient) and *banish* (patient). *Excelsior* arrives at its story-based interpretation of a user input by seeking out a path in the causal graph between the vertices provided in the input. This pathway, a sequence of connected *Scéalextric* actions, is first converted into ScéalXML and then into ComiXML. For instance, the input "My boss is a tyrant" is mapped to a path linking *command*(agt, pnt) to *despise*(pnt, agt), which is summarized by the chatbot as: "A BOSS COMMANDS, BUT TAUNTS, SO EVENTUALLY IS DESPISED BY A HARSE CRITIC." To produce a three-panel strip from its narrative, the system chooses *Scéalextric* characters to portray its key roles, such as *Boudicca* & *Spartacus* or *Sideshow Bob* & *Lisa Simpson*. Notice how *Excelsior* infers that the user despises his boss, because the boss goes from giving orders to issuing taunts.

The initiative now passes back to the user, who can either accept the system's inferences with an affirming "yes" or "OK", or reject them with a "no" or a "try again." The user can also request a near alternative, by replying "almost" or "not quite", or require that the main roles be swapped by responding "the other way around." The user can elaborate an acceptable response by replying "so?" or "then what?", or introduce a dramatic kink to the tale with a "but." Each such reply prompts the system to add another action to the developing story, and another panel to the growing comic. New additions are in turn subject to acceptance or rejection by the user, who guides the system through the story space from the initial input to the final narrative and comic strip.

This co-creative, dialogue-driven variant of *Excelsior* has yet to be fully evaluated. However, its lexicon plays a key role in the web-scale application that we will evaluate next.

[1] https://tinyurl.com/ykdn33r3 and https://tinyurl.com/4b5ekc2y

Interventions: Comics With A Purpose

We can distinguish between comics that have a meaningful narrative and those whose narrative serves a larger meaning. The former tell a story that may divert and entertain us, but the latter carry a message that their creators need us to hear. Think of the difference between the newspaper comics that avoid politics and those, like Gary Trudeau's *Doonesbury*, that weave a political stance into the subtext of their stories. Comics can be used as a candy-coloured wrapper for views that some readers deem unpalatable, thereby increasing the diversity of the audience for those views. Or, as argued by Johnson *et al*. (2020) in the context of the rancorous online debate about vaccination, by increasing the heterogeneity of arguments in a community we can stop it from growing into a self-reinforcing echo-chamber. So, as a first step to using comics as our medium of intervention into early-stage echo chambers, let's explore this debate as it unfolds on Twitter.

Characterising the data and the debate

Twitter's streaming API was used to collate a dataset of all tweets that use a relevant hashtag from a list of approx. 60 tags, from *#GetVaccinated* to *#NoVaccinePassports*. During Q4 of 2021, as national vaccination drives ran at full tilt, a corpus of 1.6M tweets from nearly 400k users was gathered.

To characterize the debate, we take a rather simple view of a user's "stance" on vaccination, and assume that one is either *pro-* or *anti*-vaccines. The debate is more subtle than this dichotomy allows, and encompasses those who chafe at vaccine mandates, vaccine passports, masks, lockdowns, or at any curtailment of their pre-Covid lives. Nonetheless, there is still a sufficient consistency of attitudes to vaccines to make this pro/anti split a useful one. To assign a stance to each user, we build a graph of all retweets in the dataset, to indicate who retweets whom, and how often they do so. In this graph we identify the 100 most retweeted users, the so-called *influencers* or *evangelists*, and manually assign a pro (+1.0) or anti (-1.0) numeric stance to each. For every other user in the graph, we now estimate their stance as the weighted average of the stances of those that they retweet, weighted by the number of times they retweet them. After 50 iterations, the initial evangelist stances percolate down to every reachable user in the graph, to mark their position in the debate as a number between -1.0 and +1.0. In total, 149,162 users (38%) are assigned a positive *pro* stance and 214,066 users (55%) are assigned a negative *anti* stance. The remaining 7% are not in this retweet graph, or are not connected to one of our initial evangelists. Of those that are assigned a stance, the pro/anti split in the debate is 41/59%.

The dataset contains 39,366 distinct hashtags, many of which show a clear pro- or anti- bias, from *#VaccinesWork* to *#DoNotComply*. To evaluate our automatic assignment of stances, we look at the most frequently-used tags in the data and identify 100 clearly *pro* and 100 clearly *anti* tags. Looking at those who use these tags, we find clear support for our approach to stance allocation. The probability that a user who uses more pro-tags than anti-tags is assigned a pro-stance is .994, while the probability that one who uses more anti- than pro-tags is assigned an anti-stance is .999.

Hashtags condense arguments into compact, meme-like forms, such as *#VaccinesSaveLives* and *#FauciLied*, so to take the pulse of a debate we must get a handle on its tags. We first assign a stance to every hashtag, by estimating the stance of each as the weighted mean of the stances of those who use it, weighted by the number of times they use it. As a result, 39% of tags are assigned a positive *pro* stance, and 61% of tags are assigned a negative *anti* stance. These tags are viewed as discrete wholes, yet most tags are multiword forms with a headline-like structure. To unzip each hashtag into its headline content, we apply the *camel case* heuristic (as interior words tend to be capitalized) and a large lexicon (specifically, the *Figaro* lexicon of the previous section) to produce a sequence of words from each composite tag. 66% of the tag set, or 25,999 tags, can be segmented in this way.

Because Figaro's lexicon supports metaphorical analysis, it categorizes its entries along image-schematic dimensions such as *accepting* (up, love, praise, pick, etc.) and *rejecting* (down, fire, kill, dump, etc.). It also marks *negation* words (no, never, don't, etc.), and those related to *responsibilities* (e.g., rule, law, tax) and *rights* (e.g., freedom, truth, choice). We extend this lexicon to also mark geographic locations, such as Australia, the USA, China, Europe and their cities. We also mark references to the left or right of the political spectrum (e.g. Dems, Biden on *left*, GOP, Trump on *right*). This allows us to characterize the argument carried in a tag along these dimensions (e.g., responsibilities in Europe, or an acceptance of rights in the USA). If we now view users as aggregates of the arguments they use, i.e. as aggregates as the hashtags they use, we can apply this characterization to users too. For instance, we can say whether a given user is more accepting than rejecting (i.e., uses more accepting arguments than rejecting ones), or more focused on rights over responsibilities (i.e. uses more tags focused on rights), or more likely to reference the political left than the right.

Unsurprisingly, the hashtags of the anti-vaccination side show a clear preference for overt negation. The probability that a negated tag is assigned an anti-stance is .762, while a negated tag that highlights responsibilities has a probability of .981 of being labeled *anti*. Similarly, any tag that shows a rejecting attitude to responsibilities has a .942 probability of being labeled *anti* (these results are significant at the $p < .0001$ level). A clear political faultline is also evident at the hashtag level. Hashtags that mix rejection and a reference to the political left have a .957 probability of being labeled *anti*, while those that mix rejection and a reference to the political right have a .920 probability of being labeled *pro*.

These regularities also hold at the user level. A user that is more accepting than rejecting, and uses more accepting arguments than rejecting ones, has a .898 probability of being labeled *pro*, while one that is more rejecting than accepting has a .915 probability of being labeled *anti*. This simple criterion accounts for 75% of all stanced users. The battlelines are very clearly drawn in this debate, since any user that makes even a single rejecting reference to the idea of responsibility (177,319 do, or 45% of all users) has a probability of .966 of being labeled *anti*. Once again, each of these findings is significant at the $p < .0001$ level.

Evaluation: Data-Driven Comics

Excelsior has been adapted to work with the outputs of the *Scéalextric* story-generator, but can it be used to represent the cut and thrust of arguments in the vaccination domain? To test its applicability to online debates, a sample of 1,500 tweets is selected at random from our vaccine dataset. Each tweet has one or more hashtags with a quantifiable stance, and each contains one or more well-formed sentences that can be used as its narrative content. The sample has an even split of *pro* and *anti* sentiments (750 tweets of each stance). To test the expressive range of the Excelsior representation, we consider whether humans versed in this representation can effectively map these 1,500 tweets into a comics form. If so, we can have faith that the representation is expressive enough for generating comics about wide-ranging concerns. Moreover, the resulting mapping provides a parallel corpus for training an automatic translator of tweets into comics – this is a subject for future work and another paper – and as a baseline for evaluating comics generated from hashtags.

Annotators are first familiarized with Excelsior's assets and its markup conventions. A lightweight markup is used in lieu of full ComiXML, so that annotators need not create detailed XML forms. Rather, the markup simply segments each text into separate panels, and identifies the figure that speaks (or thinks) each chunk of text. For each figure in the panel, a pose and an orientation is also defined, and for the panel itself, a backdrop asset may also be specified. This is sufficient for a machine to construct the full XML for itself.

All 1,500 tweets were translated into a comics form, and none were discarded or labeled as too difficult to translate. The annotators used 95% of the available assets to markup the sample, which suggests they have a broad applicability. For the sample as a whole, the 5 most frequently used pose assets are: *operating* (a figure in a surgical mask carries a syringe and knife); *experimenting* (the same figure, without the mask, cackles as a rat scurries underfoot); *defensive* (an anxious figure retreats with its arms outstretched); *running away* (a scared figure flees in terror); and *rude* (an angry figure "flips the finger"). The 5 backdrop assets most often used are: *hospital* (interior); *hospital* (exterior); *graveyard* (tombstones and grass); *government* (a view of congress); and *battlefield* (barbed wire, ruins and scorched earth).

Each tweet text to be annotated contains, on average, 14 words. When annotators segment these texts into a series of comics panels, the mean number of panels per tweet is 2.82 (sd.=.92). Most comics are thus two to four panels in length. The mean number of words per text segment – and thus, per panel – is 4.68 (sd.=2.85). Most text balloons will thus contain between two and eight words apiece. Specific assets are favoured for the depiction of the arguments from opposing stances. Vaccination is generally depicted using the *operating* pose for the "pro" tweets, and depicted using the more sinister *experimenting* pose for the "anti" tweets. The *graveyard* and *hospital* backdrops find equal favour in pro and anti tweets – the graveyard is a terminus for those who refuse vaccination, or who die from its side effects – but *government* and *battlefield* are preferred by those who campaign against (rather than for) vaccination mandates.

Hashtag Comics: Automated Generation

We estimate that 120 person-hours of effort were needed to translate this sample of 1,500 tweets into ComiXML. This investment of time and effort might be repaid by a machine learning approach to generation that uses this tagged dataset for supervised training, but it is still a significant outlay for each new debate topic to be modeled (e.g., climate change). Any automated approach to *argument-by-comics* will likely lack the finesse of a human-led one, but it should be easier to adapt to new debate topics. Whole tweets are too loosely structured to serve as the basis of this automated approach, but hashtags – which work best when used as hooks for the content they adorn – are ideal: they capture the gist of an argument in a pithy form that one hopes will "go viral."

To begin, we pass every hashtag in the dataset through a segmenter, to identify its individual words. As noted earlier, 25,999 tags (or 66% of the total) can be segmented using a lexicon and the norms of "camel casing." The latter allows a tag to be segmented even if some of its words are not in the lexicon, such as "#*CovidIsAHoax*. Figaro's lexicon will cover most of the domain-independent terms that are used, such as "is" and "hoax", but important terms like "Covid" must also be added, as must the names of central figures in the debate, such as "Fauci," "Pfizer" and "Biden." To plug the largest holes in the lexicon, we rank the unlisted terms by frequency in the tag set, and add entries for the top 200. This leads us to add entries for "Covid" and "Covid19", as well as for casual references to vaccines ("vax", "vaxxed") and for recurring characters in the debate (Tony Fauci, Joe Biden, Scott Morrison, Boris Johnson, etc.) and the various ways in which they are named in a tag. For these characters we define Excelsior specifications for their cartoon effigies (e.g., white hair for Fauci, blond for Boris, bald for Biden), and for "Covid" we specify the poses *stalking* and *preying*. Every comic hero needs a comic villain, and named figures are convenient bogeymen (or bogeywomen) for the debate. Any user that mentions *Fauci* in even a single tweet is very likely to hold anti-vax views (probability = .914), while for *Bill Gates* that probability rises to .947; for *Jacinda Ardern*, the prime minister of New Zealand, it rises to .996.

A hashtag can be automatically translated into ComiXML if: every word in the segmented hashtag has a lexical entry; the hashtag provides enough material for at least one panel; its lexical entries specify poses for two figures in each one; and, for certain terms, suggests a backdrop for the panel too. The hashtag may contain lexical items that do not translate into any visual element, such as function words ("a", "the", etc.), as these can contribute to the content of text balloons, but it may not contain a term that falls outside the lexicon. These restrictions lead to 9,802 hashtags (or one third of all tags that can be segmented) being mapped into ComiXML. In turn, 36% of these comics are produced for hashtags that are associated with a pro-stance, and 64% are generated for tags that suggest an anti-stance. For this dataset, there is a political dimension to how tags convey stances – *anti* tags lean right and *pro* tags lean left – so the translator uses this finding to colour the figures in its panels. For a comic that is created for a *pro* tag, the protagonist – the figure who utters the words of the tag – is blue, and the antagonist is red. For an *anti*-leaning tag, the comic's protagonist is red and the antagonist is blue. In the comic of Fig. 3, for the *anti* tag *#DemocratsAreDestroyingAmerica,* "Democrats" is defined by the lexicon as *one who votes for Biden*, so the red protagonist is instead paired with a cartoon Joe Biden:

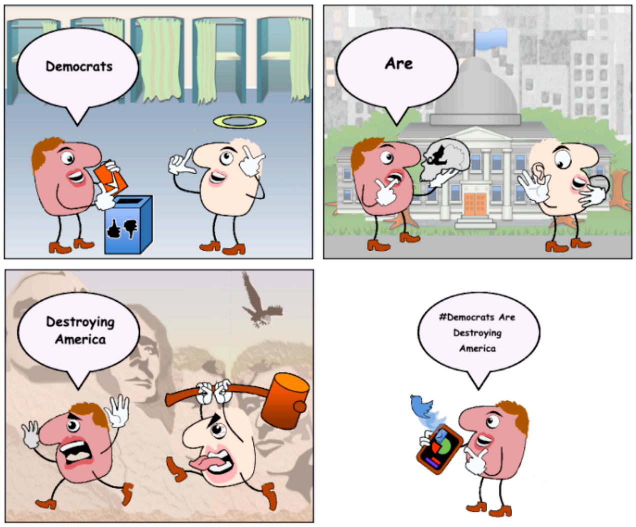


Fig. 3. Comic for the *anti* tag *#DemocratsAreDestroyingAmerica*.

As a comic unfurls from left-to-right, it reveals successive words in the hashtag. Each panel hinges on a single hashtag term that suggests a pairing of figure poses. "Democrats," for instance, suggests *voting for* and *saintly*, where the latter is the pose struck by the cartoon Biden. The setting for this panel is suggested by the *voting for* pose, which is strongly linked to backdrop *polling station* and only weakly tied to the backdrop *government*. This weak association suggests a scene change in the second panel, which stages the copula term "are" in a dramatic, Hamlet-like fashion. In the third panel, the action term "destroying" suggests a pairing of the poses *destructive* and *running away*, while "America" is rendered in the same panel as the backdrop *Mt. Rushmore*. The narrative impetus of the comic, the original hashtag, is ultimately summarized with a final borderless, blank panel.

When a hashtag takes a dramatic turn, its comic does too. By forcing the narrative into a two-fisted tale of protagonist vs. antagonist, the red protagonist becomes both Democrat and anti-Democrat, a figure that starts the tale by voting for Biden and ends it by fleeing from him in terror. We see just as dramatic an arc in the comic of Fig. 4, which is derived from the pro-leaning tag *#CovidIsNotGone*. "Covid" ranks high in the list of lexicon gaps for the vaccine dataset, and we plug the gap with an entry that specifies a colour (*green*) and a pose (*preying*). This makes "Covid" a state of being, a posture that can change from one panel to the next. In the first panel, the antagonist is portrayed as "Covid", preying on a scared protagonist. In the next, which depicts negation with a wag of the finger, the antagonist resumes her "pro" colouring. The emergent effect suggests that Covid can lurk within a character, waiting for a chance to express itself.

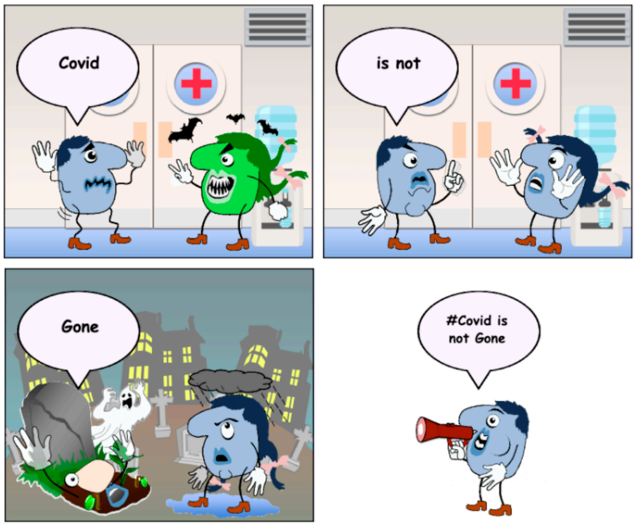


Fig. 4. A comic generated for the *pro* hashtag *#CovidIsNotGone*.

Discussion

How do the comics that are generated automatically from hashtags compare with those created manually from entire tweets? On average, a hashtag-based comic contains 2.83 panels (sd. = .687), including the final summary panel, and its average speech balloon has 1.46 words (sd. = .55). The typical hashtag comic is thus punchier than a typical tweet-based comic, yet the tag comics hit the same high notes. As was the case in the human-translated sample, this larger set uses 95% of Excelsior's assets, and likewise numbers the backdrops *hospital interior* and *government*, and the poses *operating* and *defensive*, among its top 10 most-used assets. Other highly ranked assets include the poses *preying*, *sick* and *scared* (to depict coronavirus and its victims), and the backdrops *graveyard* (for pro- and anti- deaths) and *police state* (a dark variant of *government*, to suggest oppression).

Most of Excelsior's poses depict highly emotional states, bound up in the character actions that evoke those states. A significant proportion of Excelsior's backdrops also evoke – or stand in for – emotional states, such as the *graveyard*, *police state*, *dungeon*, and *battlefield* backgrounds, which can be used literally – in a *Scéalextric* tale, for instance – or metaphorically, in debate spaces such as that for vaccines. Our sample of 1,500 pro- and anti-vaccine tweets serves as a representative slice of the Twitter debate, and since these are mapped onto Excelsior's emotional images by hand, we can be confident that these choices capture the tenor of the online debate. But how well do our hashtag comics capture the dominant themes and emotions of the debate? To assess how closely our automated comics hit the same notes, we measure the correlation between the usage frequencies of each asset for both manual and automatic comic generation.

Comparing the usage frequencies of all Excelsior assets across both sets of comics, we find that *Pearson's r* = .824. This reflects a strong correlation between the choices that humans make when visualizing online arguments, and those that the machine makes when visualizing them for itself.

Concluding Remarks

Comics are an appealing delivery mechanism and framing device for our stories, wherever they happen to come from. Data-driven comics are a specific form of “creativity with a purpose” (as opposed to “art for art’s sake”) that uses the expressive and representational affordances of the medium to convey a specific message and advance a particular goal. So, on the back of some large-scale data analysis and some small-scale plugging of lexical gaps, a comics generator can be adapted to tell the stories of a diverse body of users, and in doing so affect the ways in which they interact. That, at least, is our goal here: to tell the stories, and express the concerns, of different users to a wider audience than they might otherwise reach, and thus increase the heterogeneity of viewpoints within their opinion “silos.” We have shown how individual arguments, as expressed in individual tags, can be translated into comics without loss of emotionality. A machine can understand those arguments, like those who make them, at a general level e.g., as rejecting or accepting of particular people, policies or ideas. But, as a next step, we must do more than echo the arguments in comics form, and target them at communities that are least open to them. These practical interventions will pose the best test of the formats, resources and tools that we have presented here.

Comics can be added as a medium of expression to many kinds of computational creativity system. Story-generators are just the most obvious, and we anticipate that *ScéalXML* can be adapted – with minor extensions – to suit the needs of systems other than our own choice, *Scéalextric*. In cases where this necessitates the creation of new assets, such as pre-Columbian imagery for the *Mexica* stories of Pérez y Pérez (2007), these additions will benefit both systems. We have shown here how comics can serve as a means of *data visualization* i.e., as a means of emotively visualizing the drama inherent in aspects of a large data set. A comic can be the main creative output of a system, or a useful means of framing the system’s explanations of its own creativity. If used for the latter, we expect to find a great many outlets for comics-based creativity, in a wide range of applications that go beyond what we typically think of as story-centred.

Author Contributions

The reported research is wholly the work of the sole author.

Acknowledgements

The author acknowledges the intangible yet invaluable outcomes of his many discussions with colleagues and friends.

References

Agrawal, H., Chandrasekaran, A., Batra, D., Parikh, D., & M. Bansal. (2016). Sort story: Sorting jumbled images and captions into stories. *Proc. of Empirical Methods in NLP*, 925–931.

Alves, T., McMichael, A., Simões, A., Vala, M., Paiva, A. & Aylett, R. (2007). Comics2D: Describing and creating comics from story-based applications with autonomous characters. *In Proceedings of CASA, the 20th Annual Conference on Computer Animation and Social Agents*. Hasselt, Belgium.

Cavazza, M., Charles, F., & Mead, S. J. (2003). Interactive Storytelling: From AI Experiment to New Media. *Proceedings of the 2nd international conference on Entertainment computing*, 1-8.

Cohn, N.. (2013). *The Visual Language of Comics: Introduction to the Structure and Cognition of Sequential Images.* London: Bloomsbury.

Cohn, N. & Magliano, J.P. (2020). Visual Narrative Research: An Emerging Field in Cognitive Science. *Topics in Cognitive Science,* 12:197–223.

Eisner, W. (1985). Comics & Sequential Art. Tamarac, Florida: Poorhouse Press.

Forceville, C., Veale, T., & Feyaerts, K. (2010). Balloonics: The visuals of balloons in comics. *The rise and reason of comics and graphic literature: Critical essays on the form*. J. Goggin & D. Hassler-Forest (Eds.). Jefferson NC: McFarland, pp 56–73.

Gaiman, N. (2021). How To Create A Comic Book. *Masterclass course.* https://www.masterclass.com/

Gervás, P. (2014). Composing narrative discourse for stories of many characters: A case study over a chess game. *Literary and Linguistic Computing*, 29(4):511–531.

Iyyer, M., Manjunatha, V., Guha, A., Vyas, Y., Boyd-Graber, J., Daumé, H., III & Davis, L. (2017). The amazing mysteries of the gutter: Drawing inferences between panels in comic book narratives. *Proc. of Computer Vision & Pattern Recognition*, 6478–87.

Johnson, N.F., Velásquez, N., Restrepo, N.J., Leahy, R., Gabriel, R., El Oud, S., Zheng, M., Manrique, P., Wuchty, S. & Lupu, Y. (2020). The online competition between pro- and anti-vaccination views. *Nature***,** 582:230–233.

Kurlander, D., Skelly, T. & Salesin, D. (1996). *Comic Chat*. In proceedings of SIGGRAPH'96, the 23rd annual conference on Computer graphics and interactive techniques, pp 225–236.

McCloud, S. (1993). *Understanding Comics: The Invisible Art.* New York: Harper Collins.

McIntosh, J. (2005). ComicsML: A proposed simple markup language for online comics. *https://jmac.org/projects/comics_ml/*

Melistas, T., Siglidis, Y., Kalogiannis, F. & Manouach, I. (2021). A Deep Learning Pipeline for the Synthesis of Graphic Novels. In *Proc. of ICCC’21, the 12th International Conf. on Computational Creativity*, Mexico City.

Montfort, N., Pérez y Pérez, R., Harrell, F. & Campana, A. (2013). Slant: A blackboard system to generate plot, figuration, and narrative discourse aspects of stories. In *Proc. of the 4th International Conf. on Computational Creativity*. Sidney, Australia, June 12-14.

Pérez y Pérez, R. (2007). Employing emotions to drive plot generation in a computer-based storyteller. *Cognitive Systems Research* 8(2):89-109.

Veale, T. (2017). Déjà Vu All Over Again: On the Creative Value of Familiar Elements in the Telling of Original Tales. In Proc. of *ICCC 2017, the 8th Int. Conf. on Comput. Creativity, Atlanta*.

Walsh, J. (2012). Comic book markup language: An introduction and rationale. *Digital humanities quarterly* 6(1).

Generating Plotlines about Attempting to Avert Disasters
Paper type: Short Paper

Pablo Gervás
Facultad de Informática
Universidad Complutense de Madrid
Madrid, 28040, Spain
pgervas@ucm.es

Abstract

Narratives about disasters triggered by a chain of connected incidents need to combine very diverse elements, such as the set of possible problematic events, the causal relations between these events and their potential consequences, the set of solutions that might be applied, the relations of the solutions to the problems and the expected effects of the solutions. Successful modeling of the search space of such elements would provide means for generating plot lines based on attempts to avert disasters. A prototype has been constructed that combines features to model causality, set sequences of actions to mirror emergency protocols, probabilistic information on event sequencing and timed injection of events. The relative merits of the prototype are discussed and conclusions concerning the general task of modeling this type of narrative are drawn.

Introduction

Plot lines where a small incident sets off a chain of events that can potentially lead to catastrophic consequences are a staple ingredient of entertainment fiction. They are usually combined with additional plot lines outlining attempts to break the chain of causality leading to disaster. Of these attempts, all but the very last one will usually fail, taking the story to the brink of disaster before success is achieved.

In order to obtain a computational model that captures the characteristics of the search space with sufficient detail, the following features need to be modeled: 1) interest (events corresponding to normal/acceptable operation are not of interest to the narrative), 2) causality (of the kind that captures the relationship between certain events and their consequences), 3) projection of current events forward into time (in order to foresee unwanted developments and plan against them) 4) potential solutions (that might be applied to stop the problems).

The present paper purposefully sets out to develop a model designed to capture these aspects of the problem. Although we assume that accurate modeling of all the aspects is beyond the scope of a simple paper, we will consider an initial approximation capable of creating narratives that are both realistic and interesting.

Previous Work

A number of aspects are relevant to the modeling of these types of narrative: their relation to simulations of the physical systems involved, their adversarial nature – disaster waiting to happen vs. characters attempting to avert it – and the role of causality in them.

The construction of computer simulations of nuclear power plants as physical models (Lu 1999) has been exploited as a platform for inexpensive cognitive psychology research (Ulrich et al. 2017). However, most of the events in such simulations are likely to correspond to regular performance of the plant, and unlikely to merit inclusion in a narrative that aims to be interesting. A model aimed at generating interesting narratives needs to focus on events that fall outside the regular performance of the plant.

The task of responding to emergencies is reactive: initiating incidents trigger chains of events that put the plant and emergency responders set in motion plans to avert disaster. This opposition has been modeled in computational narrative before (Dehn 1989) in terms of authors aiming to thwart some of the goals of their characters as a means to create interest in the plot.

Models of causality based on defining preconditions for actions to represent causal relations have been used successfully to model narrative in terms of planning problems (Riedl 2004). An initial situation and a desired goal are provided and the planner finds a sequence of events to connect them. Adversarial conditions are considered in interactive narrative (Porteous, Cavazza, and Charles 2010), where an initial plan by the system may be altered by the actions of the user and the system replans periodically to ensure its final goal is always achieved.

Generating Plotlines based on Averting an Imminent Disaster

The present paper aims to develop a procedure for constructing plot lines built around the idea of an upcoming disaster set in motion by a small incident, and a series of attempts to prevent it. To develop a system capable of modelling this type of behaviour, the following assumptions are made: 1) uneventful operation of the system warrants no narrative, 2) some kind of triggering incident (that constitutes a problem or has the potential to lead to a problem) sets a narrative in motion 3) a sequence of events linked by causality is set in motion, such that 4) (at least) the final event in that sequence constitutes a problem; 5) the problem triggers a (number of) reactions, and 6) the reactions tend to be plans (possibly more than one) all designed to solve the problem (or some part of it) in some way, 7) reactions may themselves trigger additional causal chains of events and 8) further incidents (also problematic) may take place that break the causal chains arising from reactions.

Disaster movies usually combine a large number of types of events (meteorology, natural disasters, human error, terrorism, mismanagement…). Exhaustive modelling of all these types of events is beyond the scope of this paper. In order to provide a relatively simple domain with the desired complexity where causality relations between events can be defined in objective terms an existing knowledge-base for accidents in nuclear power stations[1] is chosen as underlying world model.

Traditional planning systems are ill suited for modelling this type of problem because: 1) the model cannot be deterministic (problems may or may not happen in different runs of the system, and these changes in behaviour need to be achieved without having to change the rules that govern the causality of the system), 2) emergency protocols take the form of timed sequences of events (responders are not necessarily aware of the causal chains that they are intended to trigger), and 3) potential chains of causality initially identified by the system may need to be broken by further problematic events (this allows the capture of mismatches between expectations and real system reactions).

Customised Representation of Narrative

Events are represented as atomic propositions. The causal chaining of events is modeled by the possibility of assigning to each event a set of *preconditions* that may trigger it (the activation of the preconditions is said to cause the following events). A description of the lexicon entries for events concerning how heat affects people are listed in Table 1, together with an example of how they would be chained by the system into a causal sequences of events (in the absence of interventions in the form of emergency responses).

Time is modelled in a step fashion, with a succession of turns taking place, and a set of events taking place simultaneously at each turn. The set of events that take place in a given turn are said to be *activated* on that turn.

An initial requirement is that the system allow for both user-driven *configurable* mode – to allow narratives to be generated on demand for specific situations – and an *autonomous* generation mode – where the system generates narratives by exploring the search space of possible narratives with no guidance from the user. To allow for this possibility, the system includes functionality for injecting events at particular moments in time. An *injection schedule* indicates the relative timing for a given set of events, starting from the point at which the injection is triggered. Table 2 shows an example of an injection schedule.

The second basic requirement is that the system be capable of representing the occurrence of problems that may lead to consequences, most of them undesirable. The occurrence of problems may be represented using the functionality for injecting events.

Whenever an event is activated, the system uses causal chaining basedon preconditions to identify any events that may be triggered as consequences. The full chain of consequences of an event is computed during the turn when the event is activated, but they do not start to be activated until the following turn. At each turn, the next level of consequences from preceding events is activated unless some solution has blocked them.

Table 3 presents a set of causal chains produced by the system that capture various possible problems modelled for nuclear power plants and the outcomes predicted if no action is taken.

Modelling Emergency Response to Problems

To model responses to problems, the system contemplates a set of elaborate responses (emergency plans) to problematic events (these elaborate responses are represented as patterns of actions to be undertaken in particular order and following an established relative timing between them).

Emergency responses are encoded in the system at three different levels. At the first level, a solution for a given problem associates the problem itself with the name of a particular plan of actions. At the second level, the actual plans to be carried out are represented as a timed sequence of events to be injected into the system. These timed sequences are also represented as injection schedules. At the third level, the system models consequences of actions taken as part of plans in the same way that it modelled consequences of problems. This allows these emergency plans to be expanded into causal chains triggered by actions in plans without having to explicitly list the consequences in the description of the plan. Causal links between plan actions and further events are modelled in the lexicon. Table 4 shows an example of a causal chain arising from a plan. This example shows how the problem chain shown in Table 1 may be interrupted by the application of an emergency plan.

Probability Driven Construction of Event Consequences

Causal relations between events are not always completely deterministic. Often consequences follow the events that trigger them only in some cases. This peculiarity should also be included in the model.

The system considers that, for each event that is activated, the set of its possible consequences needs to be compiled. For the events in this set, a conditional probability of occurrence given the triggering event must be considered. To inform this process, a *probability of occurrence* is associated with each event. For a more realistic model, the conditional probabilities of each consequence given its trigger should be considered, but a single probability is considered an acceptable approximation for an easier initial model.

At each turn, the consequences of events activated in the preceding turn are considered for activation. Based on

[1] Currently under development for the ADARVE project, ref. SUBV-20/2021, funded by the Spanish Nuclear Security Council.

Trigger	Event
(injected event)	HeatReachesPeople
HeatStartAffectingPeople CAUSE [HeatReachesPeople]	HeatStartAffectingPeople
PeopleSufferFromHeat CAUSE [HeatStartAffectingPeople]	PeopleSufferFromHeat
PassOutFromHeat CAUSE [PeopleSufferFromHeat]	PassOutFromHeat
PeopleDie CAUSE [PassOutFromHeat,SufferTerminalRadiationPoisoning]	PeopleDie

Table 1: Extract of lexicon entries for events concerned with the effect of heat on people (first column), together with an example causal sequence produce by the system.

Time offset	Event
0	Tornado
1	DamageToGenerator
2	NoFuelForDieselPoweredElements

Table 2: Example of injection schedule showing a tornado, damage to a generator, and lack of fuel.

DamageToDieselPumps
DieselPumpsNotWorking
NuclearReactorStartsOverheating
OverheatsNuclearReactor
NuclearReactorReachesDangerousTemperature CoolantEvaporates
HeatReachesPeople NuclearReactorStartsToMelt RadioactiveMaterialExposed
HeatStartAffectingPeople NuclearReactorMeltdown
PeopleSufferFromHeat
PassOutFromHeat
PeopleDie

Table 3: Examples of causal chains for problems in a nuclear power plants and the outcomes predicted if no action is taken. Horizontal lines represent turn transitions.

Problem/HeatReachesPeople
Problem/HeatStartAffectingPeople StartingPlanToSolve/HeatReachesPeople Solution/RemovePeopleToSafety
Problem/PeopleSufferFromHeat Solution/PeopleSafe
PlanSuceeded@HeatReachesPeople

Table 4: Causal chain arising from the activation of an emergency plan.

a random factor, consequences in this set are activated if a randomly generated number falls under the probability threshould associate to the event.

There is another aspect of consequence that needs to be captured in the model. In certain cases, two contrasting potential consequences of the same event are registered, and only one of them should be activated in any particular situation (*disjunctive branching*). To capture this feature, situations of this type are also explicitly encoded in the system and a probability is assigned to them that is used by the system to select only one of them when expanding.

The introduction of probabilistic information becomes a useful tool for customising system performance to meet the different required modes. For the *autonomous* mode, weighted random choice informed by the recorded probabilities allows for consistent variation in the outcomes. Explicit customisation of the relative values of the probabilities allows the user to constrain the search for narratives to particular subsets of the search space, allowing operation in the *configurable* mode.

Disjunctive branching is used to capture the dichotomy between success and failure of actions undertaken in pursuit of particular goals in response to emergencies. In this way, when the system is being run in the *autonomous* mode, plans may succeed or fail regardless of diligent application of established protocols, making the narratives more interesting. For the *configurable* mode, the user may tailor the probabilities used for disjunctive branches to drive system outcomes to particular alternatives.

Compilation of a Full Plot

The system operates on an injection schedule taken as input, that determines a certain sequence of events that happen at particular times. These are the incidents that create the cascade of problems. The system progressively compiles chains of consequences for events at a given turn and determines which events in those chains will happen at the next turn. It also compiles which responses may be undertaken to break the chains of undesirable consequences. Table 5 shows an example of a plot line generated for a combination of a damaged transformer and lack of fuel for diesel generators.

Due to its reliance of probabilities to drive the final outcome, subsequent runs of the system will produce plots outlines for different narratives for a given input. Additionally, the probabilities may be tailored The system can be used as a co-creation assistant, allowing the user to compile a set of plot outlines for a given configuration (input + defined probabilities) or even for combinations of different inputs and values for the probabilities.

State	
0	**Problem/DamageToPowerTransformer** (injected)
1	Problem/LowPowerToOperatePlant *(from DamageToPowerTransformer)*
2	`StartingPlanToSolve/LowPowerToOperatePlant`
	Solution/ShutDownNon-vitalSystems ToSolve/@LowPowerToOperatePlant
	`PlanEnded ToSolve/@LowPowerToOperatePlant`
3	Solution/MorePowerAvailableForVitalSystems *(from ShutDownNon-vitalSystems)*
	Problem/ElectricPumpsShutDown *(from ShutDownNon-vitalSystems)*
4	`PlanSuceeded ToSolve/@LowPowerToOperatePlant`
	`StartingPlanToSolve/ElectricPumpsShutDown`
	Solution/HookUpDieselGeneratorToElectricPumps ToSolve/@ElectricPumpsShutDown
	`PlanEnded ToSolve/@ElectricPumpsShutDown`
5	Problem/OverheatsNuclearReactor *(from ElectricPumpsShutDown)*
	Problem/NoFuelForDieselPoweredElements (injected)
6	Problem/NuclearReactorReachesDangerousTemperature *(from OverheatsNuclearReactor)*
	Problem/CoolantEvaporates *(from OverheatsNuclearReactor)*
	Problem/DieselPumpsNotWorking *(from NoFuelForDieselPoweredElements)*
	`StartingPlanToSolve/NoFuelForDieselPoweredElements`
	Solution/FuelDelivery ToSolve/@NoFuelForDieselPoweredElements
	`PlanEnded ToSolve/@NoFuelForDieselPoweredElements`
7	Problem/HeatReachesPeople *(from NuclearReactorReachesDangerousTemperature)*
	Problem/NuclearReactorStartsToMelt *(from NuclearReactorReachesDangerousTemperature)*
	Problem/RadioactiveMaterialExposed *(from CoolantEvaporates)*
	Solution/StartsDieselGenerator *(from Solution/FuelDelivery)*
8	`PlanSuceeded ToSolve/@NoFuelForDieselPoweredElements`
	Problem/HeatStartAffectingPeople *(from HeatReachesPeople)*
	Problem/NuclearFuelMeltingPointReached
	Solution/StartsElectricPump *(from StartsDieselGenerator)*
	`StartingPlanToSolve/HeatReachesPeople`
	Solution/RemovePeopleToSafety ToSolve/@HeatReachesPeople
	`PlanEnded ToSolve/@HeatReachesPeople`
	`StartingPlanToSolve/RadioactiveMaterialExposed`
	Solution/DivertReactorCoolantToWastePool ToSolve/@RadioactiveMaterialExposed
	`PlanEnded ToSolve/@RadioactiveMaterialExposed`
9	`PlanSuceeded ToSolve/@ElectricPumpsShutDown`
	Solution/PeopleSafe *(from RemovePeopleToSafety)*
	`PlanSuceeded ToSolve/@HeatReachesPeople`
	Solution/FillsUpWastePool *(from DivertReactorCoolantToWastePool)*
	`PlanSuceeded ToSolve/@RadioactiveMaterialExposed`

Table 5: Example of Generated Plot. Injected events are shown in **Bold**, the relations of causality between the various elements in *Italic*, the responses in `Typewriter` font

Discussion

The prototype includes various techniques to allow it to capture the full set of requirements identified from the formative analysis of the desired type of narratives.

The procedure for construction of consequence trees from an initiating event (problem) mirrors the representation of causal relations as used in planning-based representation of narrative, but it differs from them in that the chaining applied is not goal-driven. This is because the behaviour being modelled does not correspond to intentional actions, but rather to expanding consequences of given events. As in planning solutions applied to interactive narrative, only part of each plan built is actually used as a contribution of the narrative, because events from attempted solutions may cut off consequences of problems and further incidents may block some of the solutions.

The introduction of probabilistic information to drive the process of expanding events into consequence trees allows selective exploration of the search space. Customisation of the set of probabilities for specific runs allows relative steering of the outcome narratives.

The mechanism for injecting particular events according to a fixed schedule allows configuration of the system to produce narratives triggered by specific events, or modified at particular points by specific events.

The mechanism for modelling emergency response to problems in terms of set sequences of relatively scheduled events allows the representation of known emergency protocols or accepted plans of action in each particular domain.

All these mechanisms may be useful in the context of larger storytelling systems, where a disaster-averting plot line may need to be tailored to combine well with other plot lines (romantic interest, social problem, rivalry between characters…). Plot lines of other types may be generated by completely different mechanisms (Gervás 2021), created by hand or reused from known plot schemas (Concepción, Gervás, and Méndez 2016). Automated combination of more than one plot line may be considered (Gervás, Concepción, and Méndez 2022).

The solution employed in this paper for knowledge representation is based on simple propositional atoms, each one representing an event. This solution is considerably clumsier than those used in existing attempts at modelling narrative, such as those based on planning, grammar, or case based reasoning. However, it presents the important advantage of being an acceptable approximation to all the formalisms employed in each of those paradigms. Since the problem being considered in this paper appears to require a combination of several of these techniques, it was important to allow a solution compatible with all. Refinement of the knowledge representation can be considered as further work. In that process of refinement, the specific features that characterise each of the paradigms may be considered as possible extensions for the model, but in all cases they must be evaluated for compatibility with the elements from other paradigms that have been identified as necessary.

Conclusions

The model presented in this paper captures adequately the requirements identified for narratives about averting disasters triggered as final consequences of some initial incident.

The methodology followed in the paper, basing the construction of the model on combining a set of technologies to ensure that it contemplates the main features of the problem domain, places the priority in finding an adequate solution to a real problem. This is compatible with existing theoretical analyses of the complexity of narrative that suggest that successful modeling of narrative requires inclusion of specific representational solution for the many different aspects (Gervás and León 2014) that contribute to the richness of the medium.

With respect to future work, a number of potential lines are considered. First, the refinement of the knowledge representation schema to include information on the agents participating in the events would lead to a finer grained modeling of both the problems and the solutions being considered in the narratives. Second, since the successful modeling of the narratives in the inspiring set has been shown to require features of causality, of set sequences of events, and of probabilistic information, refinements to the model may be considered based on planning, case-based reasoning, and Bayesian inference networks.

Overall, the proposed solution remains a simple initial approximation to the problem, but it serves to highlight the need to contemplate a number of techniques to capture the full spectrum of features that are found in the narratives in the inspiring set.

Author Contributions

Pablo Gervás ideated and wrote the paper alone.

Acknowledgments

This paper has been partially supported by the CANTOR project (PID2019-108927RB-I00) funded by the Spanish Ministry of Science and Innovation and the the ADARVE (Análisis de Datos de Realidad Virtual para Emergencias Radiológicas) project (SUBV-20/2021) funded by the Spanish Nuclear Security Council (Consejo de Seguridad Nuclear, CSN). Some of the ideas in the paper benefited from discussions with Carlos León, Gonzalo Méndez and Eugenio Concepción.

References

Concepción, E.; Gervás, P.; and Méndez, G. 2016. Mining knowledge in storytelling systems for narrative generation. In *CC-NLG: Computational Creativity in Natural Language Generation (@INLG2016)*. Edimburgh, UK: ACL Anthology.

Dehn, N. 1989. *Computer story-writing: the role of reconstructive and dynamic memory*. Ph.D. Dissertation, Yale University.

Gervás, P., and León, C. 2014. The need for multi-aspectual representation of narratives in modelling their creative process. In *2014 Workshop on Computational Models of Narrative*. Quebec City, Canada: Scholoss Dagstuhl OpenAccess Series in Informatics (OASIcs).

Gervás, P.; Concepción, E.; and Méndez, G. 2022. Evolutionary construction of stories that combine several plot lines. In *Computational Intelligence in Music, Sound, Art and Design – 11th International Conference, EvoMUSART 2022*. Madrid, Spain: Springer.

Gervás, P. 2021. Computational models of narrative creativity. In Machado, P.; Romero, J.; and Greenfield, G., eds., *Artificial Intelligence and the Arts: Computational Creativity, Artistic Behavior, and Tools for Creatives*. Cham: Springer International Publishing. chapter 9, 209–255.

Lu, S. 1999. Dynamic modelling and simulation of power plant systems. *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy* 213(1):7–22.

Porteous, J.; Cavazza, M.; and Charles, F. 2010. Applying planning to interactive storytelling: Narrative control using state constraints. *ACM Trans. Intell. Syst. Technol.* 1(2).

Riedl, M. 2004. *Narrative Planning: Balancing Plot and Character*. Ph.D. Dissertation, Department of Computer Science, North Carolina State University.

Ulrich, T. A.; Lew, R.; Werner, S.; and Boring, R. L. 2017. Rancor: A gamified microworld nuclear power plant simulation for engineering psychology research and process control applications. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 61(1):398–402.

2. Co-creative systems

CHASING THE WHITE RABBIT -
A case study of predicting design phases of architects by training a deep neural network with sketch recognition through a digital drawing board

Jessica Bielski, Burak Mete,
Christoph Langenhan and **Frank Petzold**
Architectural Informatics
Technical University of Munich
Munich, BY 80333 GER
{jessica.bielski, burak.mete, langenhan,
petzold}@tum.de

Viktor Eisenstadt
and **Klaus-Dieter Althoff**
Smart Data and Knowledge Services
DFKI/ University of Hildesheinm
Kaiserslautern, RP 67663 GER
{viktor.eisenstadt,
klaus-dieter.althoff}@dfki.de

Abstract

Within this paper we propose an interdisciplinary approach at the interface of computer science and architecture to predict *design phases* using a deep neural network, based on architects' hand drawings. The overall goal of the metis projects is to provide architects with appropriate *design step* suggestions using deep learning (DL) and based on semantic information of Building Information Modeling (BIM), inspired by textual autocompletion of digital keyboards on smartphones. We describe the process of our sketch protocol study and open-source software prototype developed for sketch data acquisition with a WACOM tablet and video recordings, as well as the evaluation of the sketch protocol study and the results of the recurrent neural network (RNN) with Long Short-Term Memory (LSTM) architecture, trained with the sketch data quantified through the prototype tool. The initial prediction results of the current and the consecutive *design phase* appear promising to predict with high accuracy. Our future plans include tracking the architects design process through the labyrinth of design decision making using different mental layers (e.g. *design phases*) as filters all the way to the bottom to isolate the individual mental process of a singular *design step*.

Introduction

The world population is expected to reach ten billion by 2050 with about two thirds of the population living in the urban area (United Nations 2019). In order to meet the growing demands for residential housing, architects need to be able to work faster, more sustainably and efficiently, while simultaneously increasing architectural quality. Meanwhile, Artificial intelligence (AI) has established itself in recent years as a crucial domain of computer science for industry, research, and even daily life. Likewise, Computer-Aided Architectural Design (CAAD) and digital semantic models of Building Information Modeling (BIM) became essential aspects and everyday tools of the contemporary architectural design process. However, AI cannot be seen as a leading supportive computational method in the building sector, but promises huge potential and opportunities.

The DFG funded research project metis II aims to train recurrent neural networks (RNN) to suggest further *design steps* during the early design stages in the manner of autocompletion, inspired by the mechanisms of digital keyboards on smartphones. The intelligent system generates suggestions using deep learning (DL) and case-based reasoning (CBR) methods that analyse the design process sequences found in the training set. This approach is derived from the use of reference buildings for designing architecture - as a source of inspiration, design decisions and explicit information, and a tool for evaluation and communication (Richter 2010). We attempt to assimilate to the conversational idiosyncrasies of the designer, following the idea of the 'Architecture Machine' by Negroponte (1973). Similar to an actual conversation, the intentions of the architect needs to be clarified in the interaction between the AI and the operator for progressing and suggesting. Even more so, between an architect and its supportive intelligent system, as the designers workflow can become disrupted, if questions are answered "which are not yet being addressed, ... [implying] that more is known about the solution than is really the case" (Lawson 2004, p. 53). As sketching supports the development of ideas and intentions (Lawson 2005), and is an effective tool for communication, sketch-based interaction is a promising method for an intuitive interaction with CAAD systems, naturally integrating into the design process (Leclercq and Juchmes 2002).

In this paper we present our approach for autocompletion, as well as our sketch protocol study. We describe the process for the data acquisition, analysis and pre-processing of an architectural sketch protocol study, as well as for training an RNN with our collected sketch data acquired through dividing the sketch protocol data into *relational* design sequences, such as *design phases*, *design intention* and *design steps*, to train an RNN to detect design process patterns.

Related Work

The idea of an intelligent design assistant supporting architects design decision making is derived from the 'Architecture Machine' of Negroponte (1973) and digital keyboards on smartphones. It is to support the user by predicting and offering suggestions based on architectural design knowledge we acquired through sketch protocol studies. Sketch protocol studies are a common research tool for observing architects and their design process. The protocol study types range from 'Thinking aloud' studies for simultaneous reporting and explaining by the participant to retrospective studies with reporting and retracing the steps and decisions afterwards. Suwa and Tversky (1997), as well as Lawson (2004), have found retrospective sketch studies to be more natural for the sketching participants because of an uninterrupted design flow and being more true to a genuine work environment. Further, Suwa and Tversky (1997) propose video recording the sketch process for supporting the participant during the consecutive reporting in order to avoid selective recall, as architects and designers "are notoriously good at post-hoc rationalization of their processes" (Lawson 2004, p. 16). However, neither protocol study type results in quantitative data so far, solely qualitative ones.

In order to obtain quantitative results, categorisation needs to be introduced to the rich sketch data. Thus, Lawson (2004) presents the possibilities of *temporal* or *relational* segments for sequencing sketch protocols. Nevertheless, he sees only the *relational* ones are a true possibility for creating reproducible results, without "the assumption that they are also 'capable of characterising designing"' (Lawson 2004, p. 16). Consequently, Lawson (2004, 2005) proposes the orderless *design phases* connected via 'negotiations': *Analysis*, *Synthesis*, *Evaluation* and *Communication*, which are similar to the loosely 'interrelated' phases by Laseau (2000): *Analysis*, *Exploration*, *Discovery*, *Verification* and *Communication*. The two authors differ as Laseau (2000) further divides the *Synthesis* into *Exploration* and *Discovery*, while both agree on *Communication* being a separate category that is continuously accompanying the different phases. Furthermore, Barelkowski (2013) introduces *Knowledge Management* as part of the internal *Communication* of the architect with their own ideas specifically for the *Analysis* into the design process, e.g. deliberate ignorance of certain aspects, criteria or constraints, for being able to progress within the design process of controlled convergence (Buxton 2007). Thus, Barelkowski (2013) divides the *Analysis* into *Knowing* and *Understanding*.

Such quantifiable sequencing can be used to train an AI with sketch protocol data using supervised DL models based on RNNs (Sutskever, Vinyals, and Le 2014), whereat LSTM (Hochreiter and Schmidhuber 1997) or Gated Recurrent Units (GRUs) (Cho et al. 2014) are possible underlying sequence processing technologies. Further, other parameters, e.g. time, coordinates and pressure during the hand drawing process, can be traced and quantified through frequency, similarity and amount.

Approach

In this paper we propose a novel sequence learning based approach for the detection and prediction of architects' *design phases* using quantitative data acquired through sketch protocol studies. Within the following paragraphs we present our autocompletion approach, the data acquisition and analysis of the sketch protocol study data, and the pre-processing and integration of the data into an RNN model.

For our autocompletion approach we envision a closed AI pipeline of components that recurrently inform and learn: *Quantitative Analysis*, *Training the RNN* and *Sequencing the Design Process* (see Figure 1). We draw from the field of *Human-Computer-Interaction* (HCI), specifically Human-

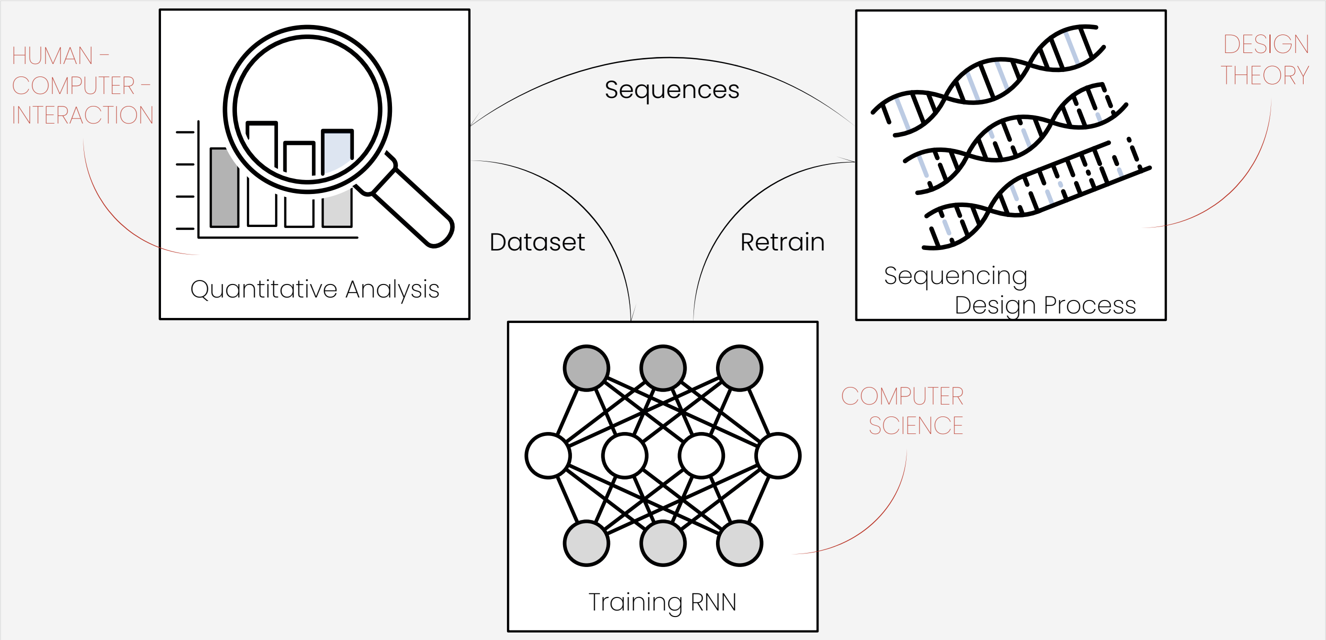


Figure 1: Envisioned autocompletion approach of the metis projects.

System-Interaction (HSI), to obtain quantifiable results of sketch protocol studies as a genuine practice of the early design stages of building design, based on sequences of the design process, found in the research field of *Design Theory*. The quantitative results of sketch protocol studies are used as a dataset to train an RNN (i.e. area of *Computer Science*), which is again used for retraining to improve the detection of sequences of the design process.

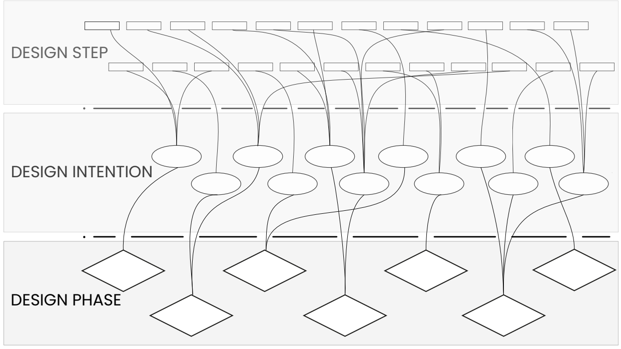


Figure 2: Visualisation of the mental layers as used for sequencing from *design step* through *design intention* to *design phase*.

We aim to track the design decision making process of architects to obtain sequences for quantifying the design process. Drawing from Lawson (2004; 2005), Laseau (2000), Barelkowski (2013), Darke (1979), and Schön and Wiggins (1992), we propose three different mental layers of relational sequences of a design decision (see Figure 2): the *design step* (e.g.'outlining parcel') as the finest clustering category, followed by the *design intention*, i.e. intention behind the executed *design step* (e.g.'requesting/requested information' by rendering the parcel dimensions tangible), culminating in the broadest sequence, *design phases*. Based on the aforementioned authors, we formulate the design process as six phases without any order, but with an overarching *Communication* to further elaborate the common *Analysis - Synthesis - Evaluation (ASE)* model: *Analysis - Knowing*, *Analysis - Understanding*, *Synthesis - Exploration*, *Synthesis - Discovery*, *Evaluation - (Informing) Knowing* and *Evaluation (- Final)* (see *Related Work* Section).

Within this paper we describe our specific approach for acquiring a first dataset for training an RNN using protocol studies, as is illustrated in Figure 3. For our study we have presented so far eight architects of different specialisation, experience level and age with the following design task:

> *A one-storey high two unit complex (52 sqm per unit), detached from the surrounding buildings, is built as a first step to extend student housing in the Olympic Village of Munich. The main facade of unit 1 faces North, while unit 2 is east-bound. One unit consists of 1 living room, 1 kitchen, 1 bathroom, and 2-3 bedrooms.*

After reading the design task accompanied by site plans and answering possible questions concerning the task, the participant draws schematic designs with a WACOM FL 0.4 pen on a piece of paper - to enable a genuine architectural design process of sketching - on top of a WACOM tablet for 15 minutes, while being video recorded. The architect can choose to place additional layers of transparent sketch paper on top or switch between any number of pieces of paper. The WACOM tablet traces the sketching, including the parameters of time, pressure and coordinates, and saves the sketch data. Afterwards the architect is being video recorded while retrospectively reporting on the design process of the sketching, while watching the previously recorded video.

After processing the video data into transcripts, one study sessions provides us with: two videos (*Sketching* and *Retrospective*), two transcripts (*Sketching* and *Retrospective*), and the *Sketch data*. To pre-process the *Sketch data* for receiving quantifiable architectural design process data, we introduce the previously described *design phases* including *Communication*, to the sketch data as sequences of the design process, as well as *architectural objects* (e.g. 'room', 'wall'). We create custom labels, which are manually assigned to the sketch protocol study data (i.e. sketch data, transcripts) (Bielski et al. 2022a), using our own open-source sketch protocol analyser tool (Ziegler 2021). The different output files in the form of JSON objects are introduced to an LSTM-based model as the consecutive RNN in our DL pipeline using the TensorFlow library (Abadi et al. 2015). The LSTM model itself includes a layer for pre-processing the quantitative aspects, namely time, pressure and coordinates, and the normalised sketch protocol data labelled with *design phases* and *architectural objects*. Based on a supervised learning scheme, the LSTM is trained with this data including temporal correlations, using a 10-fold cross validation with data samples from randomly selected time periods. The details of the mode of operation of the LSTM will be published in our paper at the ECPPM 2022 (Mete et al. 2022).

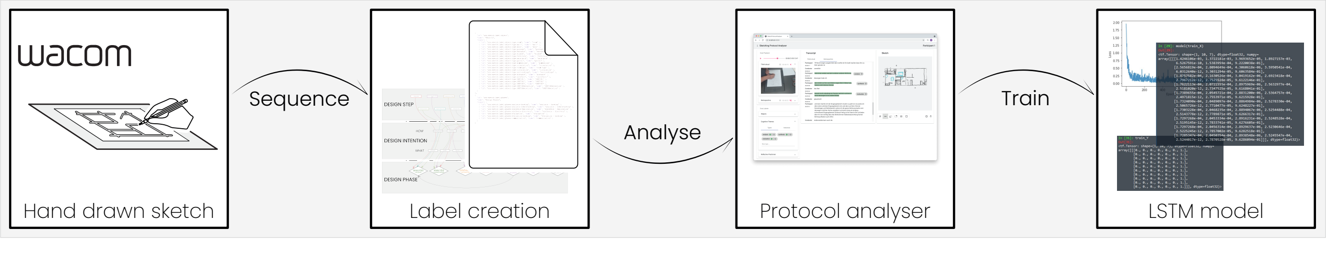


Figure 3: Process of our sketch protocol study.

Results and Discussion

The overall impact of an intelligent design assistant, suggesting further design steps, on the architect and their architectural design decision making process is to be examined for possibly hemming the creative design process and even imposing decisions.

However, our evaluation for an intelligent design assistant, suggesting further design steps enhanced with explainability, suggests that the cognitive horizon of architects is broadened by simpler and earlier access to additional information (i.e. explanation visualisations) and new perspective through other possible design solutions (i.e. design suggestions). Nevertheless, the design suggestions must be provided in a clear way as suggestions to ensure the user's ownership over the design decisions (Bielski et al. 2022b).

Further, the first results of the sketch protocol study suggest that following the design process through the mental layers of Figure 2, a domain expert can successfully assign the *design phases* to the sketch data and the transcripts for uniformly labelling the sketch protocol study data. Thus, we are able to obtain quantifiable sketch data. Furthermore, the training of our LSTM shows promising results as we are able to predict both the current and consecutive *design phase* with an accuracy of 94% (Mete et al. 2022).

The successful workflow and encouraging results need to be viewed on the background of their limitations. The researcher, an architect, labelling the sketch protocol data, has prior knowledge of *Design Theory* and analysis of the architectural design process, resulting in possible biases. Further, the amount of training data is limited (8 participants) due to a difficult recruiting process because of the COVID-19 pandemic. Finally, the *design phases* are the broadest sequencing method proposed for segmenting the architectural design process, entailing few *design phase* changes per study session: approx. 10 to 20 changes per 20,000 timestamps.

In order to remedy these shortcomings, we have taken the measures and adjustments, such as using the characteristics of the 'reflective practitioner' (Schön and Wiggins 1992) and 'primary generator' (Darke 1979) for supporting the identification of the *design phases*. To temporarily overcome the data acquisition bottleneck to properly train an LSTM, the protocol data is sliced into processing windows of fifty timestamps. Consequently, we increase the amount of data for the system to learn from and afterwards randomly separate it into training and testing data for improved data quality. Finally, in order to increase the time accuracy of the system for determining the changes of *design phases*, we define a custom loss function as an augmented version of the currently applied *binary cross entropy* loss to instead emphasise on the learning of sequence windows, which include *design phases* changes and thus, their pattern for transition.

Conclusion and Future Work

Through our sketch protocol study we explore the possibilities to track the design process through investigating the design decision making of architects using sketching on a digital drawing board for creating design autocompletion (i.e. the ultimate goal of the metis projects), as well as attempt to begin building a training set for an ANN. Our study results suggest that sketch data from sketch protocol studies can be quantified, using labels of *design phases*, derived from *Design Theory*, and our open-source sketch protocol analyser tool, based on *HCI* methods. Our *Computer Science* approach for a sequence learning based LSTM for tracking the design process by the means of these labels complements these methods to build a base training set.

So far, we have sequenced the design process of the early design stages with the broadest segmenting sequence, i.e., *design phases*. We plan to further quantitatively investigate the sketching process using the rest of the previously defined mental layers (see *Approach* Section). The next step is to consider the *design intentions* until finally, we are able to detect and predict appropriate *design steps* to suggest a continuation of the design process.

Author Contributions

Jessica Bielski (author 1) was in charge of writing the manuscript and creating the visualisations, which Burak Mete (author 2), Viktor Eisenstadt (author 3) and Christoph Langenhan (author 4) reviewed and edited. Christoph Langenhan (author 4) and Klaus-Dieter Althoff (author 6) have supervised and validated the research of the metis projects, and managed project administration in collaboration with Frank Petzold (author 5). Further, author 1 planned and conducted the study, and curated and pre-processed the collected data. Authors 1 through 4 were responsible for the conceptualisation and general methodology, whereat author 1 was responsible for the sequencing of the design process and quantitative analysis. Author 2 investigated and realised designing and training an RNN with acquired data of the sketch protocol study with author 1 and 3 as supervisors.

Acknowledgments

We want to thank the participants of the in-person study for their valuable time, feedback, and trust. Further, we want to thank the German Research Foundation (DFG) for funding the metis projects.

References

Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Goodfellow, I.; Harp, A.; Irving, G.; Isard, M.; Jia, Y.; Jozefowicz, R.; Kaiser, L.; Kudlur, M.; Levenberg, J.; Mané, D.; Monga, R.; Moore, S.; Murray, D.; Olah, C.; Schuster, M.; Shlens, J.; Steiner, B.; Sutskever, I.; Talwar, K.; Tucker, P.; Vanhoucke, V.; Vasudevan, V.; Viégas, F.; Vinyals, O.; Warden, P.; Wattenberg, M.; Wicke, M.; Yu, Y.; and Zheng, X. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Available from https://www.tensorflow.org/. [Computer Software].

Barelkowski, R. 2013. *Learning design from knowing less. On approaches to architectural design process*. LUCA Sint Lucas School of Architecture Ghent-Brussels KU Leuven Faculty of Architecture, 1st edition. chapter 8.1, 517–526.

Bielski, J.; Langenhan, C.; Ziegler, C.; Eisenstadt, V.; Dengel, A.; and Althoff, K.-D. 2022a. Quantifying the intangible - a tool for retrospective protocol studies of sketching during the early conceptual de-sign of architecture. In *International Conference of the Association for Computer-Aided Architectural Design Research in Asia*, volume 1 of *CAADRIA*, 403–411. Hong Kong: Association for Computer-Aided Architectural Design Research in Asia.

Bielski, J.; Langenhan, C.; Ziegler, C.; Eisenstadt, V.; Petzold, F.; Dengel, A.; and Althoff, K.-D. 2022b. The what why, what-if and how-to for designing architecture - explainability for auto-completion of computer-aided architectural design of floor plan layouting during the early design stages. In *International Conference of the Association for Computer-Aided Architectural Design Research in Asia*, volume 2 of *CAADRIA*, 435–444. Hong Kong: Association for Computer-Aided Architectural Design Research in Asia.

Buxton, B. 2007. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann, 1st edition.

Cho, K.; van Merrienboer, B.; Gülçehre, Ç.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Moschitti, A.; Pang, B.; and Daelemans, W., eds., *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, 1724–1734. ACL.

Darke, J. 1979. The primary generator and the design process. *Design Studies* 1(1):36–44.

Hochreiter, S., and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation* 9(8):1735–1780.

Laseau, P. 2000. *Graphic thinking for architects and designers*. Wiley, 3rd edition.

Lawson, B. 2004. *What Designers Know*. Oxford: Elsevier/Architectural Press, 1st edition.

Lawson, B. 2005. *How Designers Think, Fourth Edition: The Design Process Demystified*. Taylor & Francis Ltd, 4th edition.

Leclercq, P., and Juchmes, R. 2002. The absent interface in design engineering. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 16:219 – 227.

Mete, B.; Bielski, J.; Eisenstadt, V.; Langenhan, C.; Petzold, F.; and Althoff, K.-D. 2022. Predicting semantic building information (bim) with recurrent neural networks. In *European Conference on Product and Process Modeling*, volume - of *ECPPM*, –. Trondheim, Norway: European Association of Product and Process Modelling. [Accepted].

Negroponte, N. 1973. *The Architecture Machine: Toward a More Human Environment*. The MIT Press.

Richter, K. 2010. *Augmenting Designers' Memory - Case-Based Reasoning in der Architektur.* Ph.D. Dissertation, Bauhaus Universität Weimar.

Schön, D. A., and Wiggins, G. E. 1992. Kinds of seeing and their functions in designing. *Design Studies* 13:135–156.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In Ghahramani, Z.; Welling, M.; Cortes, C.; Lawrence, N.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Suwa, M., and Tversky, B. 1997. What do architects and students perceive in their design sketches? a protocol analysis. *Design studies* 18(4):385–403.

United Nations. 2019. *World Urbanization Prospects: The 2018 Revision (ST/ESA/SER.A/420)*. New York: United Nations.

Ziegler, C. 2021. Sketch protocol analyser. Available from https://github.com/KSD-research-group/sketch-protocol-analyser. [Computer Software].

The (Artificial) Physicality of Creativity: How Embodiment Influences Perceptions of Creativity

Caterina Moruzzi
Department of Philosophy
University of Konstanz
Konstanz, 78464 Germany
caterina.moruzzi@uni-konstanz.de

Abstract

The generation of artifacts through computational creativity (CC) systems is hitting the headlines with increasing frequency. Although impressive, this paper will not focus on the outcomes achieved by these systems, but rather on a specific dimension of artistic processes: embodiment. I discuss the results of a recent factorial survey study aimed at testing the influence that embodiment has on the evaluation of creativity. These findings show that the physical dimension of artificial systems interacting with human artists contributes to the perception of the interplay between artificial and human agents as a *creative* collaboration. I propose that a closer study of the dynamics of interaction between embodied machines, human artists, and the public can facilitate progress in both the artistic and the technology sector.

Introduction

In the last decades, computers and Artificial Intelligence (AI) systems have been increasingly involved in the field of creativity, by generating creative artifacts or by assisting human artists in their creative processes (Lubart 2005; Marks 2015): composing music in the style of Bach (Huang et al. 2019), creating paintings sold for hundreds of thousands of English pounds at renowned auction houses, and even having their say in the fashion industry (Byers 2020). The rapid technological development of AI systems and the advances in the computational creativity (CC) field demand a more detailed and overarching analysis of the impact that the deployment of technology in the arts can have on different aspects of the creative world.

This paper discusses the results of a recent study on the influence of embodiment on the perception of creativity in human and artificial systems. In the design of, and aims behind, the study, I assumed the validity of the hypothesis made by Guckelsberger et al. (2021) that "furthering our insights into embodied computational creativity (CC) will play a critical role in advancing the goals of CC more generally." With a few exceptions (Sharples 1994), the role of embodiment in creativity has arguably not been investigated in depth in the literature, and even less so in connection with AI. Still, the perception of the artists' embodiment is generally deemed to be a key aspect of the observer's response to the artwork (Freedberg and Gallese 2007). The aim of the paper and of the study here reported is, thus, to contribute to closing this gap by reporting empirical findings on the influence of embodiment on perceptions of creativity. [1]

Rather than just focusing on *artistic* creativity, the study examined perceptions of creativity also in the context of *scientific* practices. This was done in accordance with the belief that creativity is not limited to artistic practices, but should instead be investigated in a wider spectrum of fields and disciplines, including science (Pease et al. 2019).

Background and Related Works

Sensory-motor intelligence is a crucial aspect of human and animal intelligence, and a key requirement to develop common sense knowledge (Pfeifer and Iida 2004). In social science, the influence of the embodiment factor in shaping cognitive processes, is strongly advocated by the embodied mind paradigm (Foglia and Wilson 2013; Varela, Thompson, and Rosch 2017). Since the promotion of research in embodied intelligence in the Nineties by Rodney Brooks (Brooks 1991), and the arguments against cognitivism and neuro-reductionism which started to gain traction in many fields, the field of robotics has been involved in the development of AI as a discipline in an increasingly substantial way.

Robotics and embodied intelligence are employed for a wide range of tasks, from space exploration to industrial manufacturing, including applications in the creative sector. Already in the eighteenth century, the fascination for creating art through and with robots started with the creation of, among other robotic systems, the humanoid automata created by the watchmaker Pierre Jaquet-Droz (Leymarie, Bessette, and Smith 2020). Recently, the interest of both artists and computer scientists for 'creative' machines increased, for example with the creation of 'painting robots' (Cohen 1995; Deussen et al. 2012; Jean-Pierre and SaId 2012; Smith and Leymarie 2017; Srikaew et al. 1998; Tresset and Leymarie 2013; Yu and Chen 2018).

[1] The notion of embodiment that will be assumed is that of 'physical' embodiment, namely "characterizing systems with a physical body that can interact with the environment by being subjected to and by exercising physical force" (Guckelsberger et al. 2021).

The embodiment dimension introduces constraints that may not be present in purely computational settings, and these constraints may contribute to enhancing creativity (Costello and Keane 2000; Johnson-Laird 1988). Still, a general reluctance at attributing creativity to AI, irrespective of whether it is embodied or not, is well-known and addressed in the literature (Mumford and Ventura 2015; Colton 2008; Jordanous 2012; Natale and Henrickson 2022). Previous studies aimed at investigating this phenomenon, by focusing on the evaluation of perceptions of creativity (Jordanous 2012; 2016; Karimi et al. 2018). The present study inserts itself into this dialogue, contributing to the investigation of creativity attribution through empirical insights resulting from the manipulation, made possible by a factorial survey methodology, of the dimension of embodiment and other dimensions (Simo Linkola and Kantosalo 2022).

Study on the Influence of Embodiment on Perceptions of Creativity

Aims

Answering whether artificial systems can be deemed creative is not among the scopes of the study. Acknowledging the contested nature of the concept of creativity, this paper will not be trying to propose a definition of creativity, either (Jordanous 2016; Jordanous and Keller 2016; Moruzzi 2021). Rather, the discussion will focus on the influence of embodiment on evaluations of creativity.

The starting hypothesis of the study is the following:

> **Hypothesis.** Between the attribution of creativity and the embodied presence of the actor performing the process under examination, there is a positive correlation.

Namely, artificial systems possessing physical actuators through which to perform an action can be considered more creative than systems that reach the same result but with no physical intervention on the surrounding environment.

This hypothesis is motivated by some studies carried out in online and live contexts (Herman and Hwang 2022), and by past surveys conducted by the author on creativity perceptions of the process and products by generative art algorithms (Moruzzi 2020b). Participants to these surveys expressed the belief that an essential dimension for creativity is the physical presence of the artist during the creative process, a dimension that was deemed as lacking from the systems under examination.

In their overview of academic publications on embodied computational creativity in the last ten years, Guckelsberger et al. (2021) indicate some directions for future work in the field of computational creativity. In particular, they suggest to (i) "conduct qualitative and quantitative empirical studies on the impact of a specific embodiment, treated as independent variable, on (the perception of) creativity", in order to produce generalizable and empirical studies on the effect of embodiment on artificial creativity and its perception, (ii) employ objective and not subjective measures of creativity when conducting these studies, and (iii) avoid ambiguous uses of the concept of creativity. This paper responds in particular to suggestion (i), reporting the results of an empirical study conducted through online factorial survey experiments on perceptions of creativity in human and artificial agents.[2]

In addition to the exploration of the impact of the embodiment dimension, the study presented in this paper was designed also to test the influence of other dimensions on perceptions of creativity: agency, explainability, and the artificial or biological nature of the actor performing the action. In the interest of the focus of the present paper, the analysis of the influence of these other dimensions will not be addressed.[3]

Procedure

Participants were recruited online through academic newsletters in philosophy, art, and computer science. Data collection took place over three weeks in July 2021. Participation to the online questionnaire was voluntary and no information has been collected that could directly or indirectly identify a survey participant. After successful participation in the survey, respondents have been asked for their email address in a separate survey to participate in a raffle for one of three €50.00 e-commerce vouchers as an incentive for participation.

The time needed for completing the online survey was of around 15 minutes. Participants first completed an online consent form and a demographic questionnaire that included questions about their age, level of education, field of studies, and current occupation.

In the second part of the study, participants were asked questions regarding their intuitions about features of agency and creativity. Results regarding agency attribution will not be reported here as they are not relevant in respect to the focus of this paper.

Regarding creativity, respondents were presented with the question: "Which of these concepts do you associate with the notion of 'creativity'?" and they were asked to choose all the features that applied from the ones reported in the list of Table 1. These attributes were chosen among the ones that are more commonly associated to creativity in the literature on the topic (Jordanous and Keller 2016; Moruzzi 2021). In brackets is the number of the times that each attribute has been selected by respondents.

Factorial Survey Experiment

The central section of the questionnaire consisted in a factorial survey experiment, an approach which presents study participants with different vignettes which describe hypothetical scenarios (Auspurg and Hinz 2014). In this study, vignettes were in the form of a short text, but they can also be images or videos. The situations outlined in the vignettes have different attributes (dimensions) and participants are

[2] With 'agent' I understand in this paper "anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators" (Russell and Norvig 2011). I will use the term 'actor' in the instances in which I do not assume that the individual necessarily possesses agency, as the latter was one of the independent variables of the study.

[3] These other dimensions will be investigated in a paper in the proceedings of the xCoAx 2022 conference, currently in press.

Creativity Attributes
Novelty (128)
Problem-solving (87)
Surprisingness (66)
Value (52)
Instinctiveness (5)
Serendipity (22)
Unexplainability (20)
Genius (33)
Pleasantness (4)

Table 1: *Creativity.* List of creativity attributes, participants had to choose from. In brackets is the number of times each attribute has been selected.

asked to express their judgement regarding them. The values (levels) of the dimensions are varied in order to test the impact that they have on the participants' evaluation. The factorial survey design was particularly beneficial in this context, as it enables to test the effect that the manipulation of independent variables (in this case, embodiment) has on the dependent variable (in this case, creativity perception). In Table 2 are the variables that had been included in the experiment:

Variables	
Independent	Identity of the actor
	Agency
	Embodiment
	Explainability
Dependent	Agency Attribution
	Creativity Perception
	Authorship Attribution
Control	Process performed

Table 2: Independent, dependent, and control variables used in the factorial survey experiment.

Results in respect to the influence of the dimensions of Agency and Explainability on creativity will not be reported as they are not relevant to the focus of the present paper.

The process of creation, performed in the experiment, was kept constant, as control variable. The focus on the role of the body in both the creation and the appreciation of creative processes centers the discussion around creativity as a process rather than a product. While there is no doubt that machines can produce artifacts that are aesthetically appealing, more critical is the question of whether the process they undertake in order to create the latter can be deemed creative. By focusing on the process, it is possible to assess the experience behind the creation of an artifact and, thus, compare human and machines that engage with creative processes (Leymarie, Bessette, and Smith 2020).

Different vignettes resulted through the combination of the different levels of the variables, or dimensions, above mentioned. Figure 1 shows the dimensions and variables of the 8 vignettes present in this study. A random selection was programmed into the survey to determine which vignettes to present at the beginning of the survey to each respondent.

Each respondent was assigned two vignettes, constructed on the basis of two scenarios: Scenario A. Painting a canvas, and Scenario B. Discovering a vaccine. In reading the text of the vignettes, participants were asked to engage in a thought experiment. They could not actually perceive the process described in the vignette and were required instead to imagine the process and the properties involved. The following is the structure used for the vignettes. Between brackets are the dimensions, the value of which is manipulated.

Scenario A: Painting a picture

[Actor] is/are in the Royal Moondust Academy of Arts to paint a canvas. The process [actor] undertake/s is the following:
(If Displaying agency:) First, [actor] [agencyattribute1adv] selects the color palette and tools needed to paint the picture, then starts painting on the canvas. Lastly, [actor] [agencyattribute2adv] observe/s the picture and decide/s to stop painting, as the current state is the best possible result that can be obtained.
(If Not displaying agency:) First, [actor] randomly pick/s some colors and tools, then starts painting on the canvas. Lastly, [actor] all of a sudden, lift/s the brushes from the canvas and stop/s painting.]
The final painting is considered to be visually pleasing by a general audience.
(If Explainable:) A faithful record of the process of the painting of the canvas is published in an open-access journal. All the processes made to achieve the result are explicitly reported and clearly explained in a language understandable to a non-specialist audience.
(If Not explainable:) No record of the creation of this painting is available because a full report of the processes that led to the final result could not be produced by [actor].

Scenario B: Vaccine discovery

[Actor] work/s in the Research Laboratory of Sundance University to perform experiments to find a vaccine against COVID-19. The process [actor] undertake/s is the following:
(If Displaying agency:) First, [actor] [agencyattribute1adv] generate/s hypotheses from the available background knowledge and models, then carry/ies out the experiments in the Lab. Lastly, [actor] [agencyattribute2adv] analyze/s and interpret/s the results obtained.
If not displaying agency:) First, [actor] automatically tries all combinations of the available background knowledge and models to generate hypotheses, then carries out the experiments in the Lab. Lastly, Dr. Miller generates the results by performing mathematical calculations and selecting the more statistically relevant answer.
With success! Through the experiment [actor] find/s out a specific feature of the protein shell of the SARS-CoV-2 virus. This allows [actor] to develop a vaccine that is able to train the immune system to combat not only the

known variants of the virus but also every possible future mutation of it. And what's more, the vaccine works against all influenza viruses! The vaccine goes through rigorous testing and it is finally approved and licensed.
(If Explainable:) A faithful record of the experiment is published in an open-access journal. All the passages of the experiment and the processes made to achieve the result are explicitly reported and clearly explained in a language understandable to a non-specialist audience.
(If Not explainable:) No record of the experiment is available because a full report of the processes that led to the discovery could not be produced by [actor].

Participants had to read the vignettes and provide their impression of the levels of agency and creativity displayed by the actors in the presented scenarios.

In what followed, respondents were then asked to motivate their answers through a free response field, compulsory to move on with the questionnaire. Comments have been first organized in an Excel spreadsheet according to the scenario and vignette they were referring to. Within the single vignettes, they have then been arranged in descending order, according to the corresponding rating of creativity and agency that had been given by the respondent (Tables 6, 7) The content of the responses was then qualitatively analyzed using a grounded theory methodology (Charmaz 2006; Martin and Turner 1986). This method was chosen as opposed to the method applied in the analysis of the rest of the survey. Instead of starting with a hypothesis, e.g., the hypothesis on the correlation between embodiment and perceptions of creativity, the comments were analyzed without starting from any assumption, in order to test whether what emerged from the comments confirmed the results of the other sections of the survey.

Results

Demographics

The final sample consisted of 161 participants. The mean age is of 39.1 years. 157 out of 161 participants have a university-level education. 126 participants have a humanities, 22 an artistic, 15 a scientific, and 11 a technology educational background (selection was not mutually exclusive). The current occupation of most of the participants is in the education sector (Student 44, Academic 66, Engineer 3, Teacher 10, Admin 7, Retired 6, Other 25). The prevalence of participants with an educational and/or academic background and occupation is due also to the channels through which the survey has been advertised.

Factorial Survey

After carefully reading the vignettes, respondents were asked to rate the process of, respectively, the creation of a painting (Scenario A) and the discovery of a vaccine (Scenario B) for their creativity on a 7-points scale from 'Not at all creative' to 'Very creative'. In both Scenario A and Scenario B, the average creativity was evaluated at 0.6 points, slightly above the mid-point of the scale.

What is more interesting, though, is to examine how the perception of creativity is affected by the manipulation of the different dimensions. Table 3 shows how the participants' evaluation of creativity changes by varying the Actor dimension, namely by presenting the process as performed by a human, an AI, a team of a human with an AI, or a team composed of two AIs. Values are rounded to the nearest hundredth, and they are reported in respect to the baseline (0) which corresponds to an individual human actor.

Statistically significant results, i.e., when the "p-value" ($Pr(>|z|)$) is inferior to 0.05, are marked in Table 3 with an asterisk.[4] Just for the fact of not being a human, but rather an artificial actor (other dimensions being equal), the AI is judged as 0.88, 1.00 and 0.98 points less creative than an individual human actor. What may come as a surprise, is that also the Human+Human team has been judged as 0.54, 0.74, and 0.68 points less creative than an individual human actor.

	Est.	**Std. err**	**z value**	$Pr(>\|z\|)$
	Painting scenario			
Human	0	0	0	0
AI	-0.88	0.43	-2.00	0.04*
Hum.+Hum.	-0.54	0.37	-1.44	0.15
Hum.+AI	-0.18	0.38	-0.48	0.63
	Vaccine scenario			
Human	0	0	0	0
AI	-1.00	0.43	-2.31	0.02*
Hum.+Hum.	-0.74	0.37	-2.01	0.04*
Hum.+AI	-0.58	0.43	-1.36	0.17
	Combined scenarios			
Human	0	0	0	0
AI	-0.98	0.32	-3.08	0.002*
Hum.+Hum.	-0.68	0.25	-2.68	0.007*
Hum.+AI	-0.39	0.27	-1.46	0.14

Table 3: *Actor Dimension.* The table shows the impact of the manipulation of the Actor dimension on the perception of creativity.

Table 4 shows how the participants' evaluation of creativity changes by varying the Embodiment dimension in respect to the baseline, which corresponds to the actor being not embodied. Only the results where Actor = AI are reported, as it is assumed that all humans are embodied.

In both Scenario A and B, when the actor is described as embodied (i.e., as a robot acting through robotic arms), the evaluation of creativity is lower than in the case of the actor being a computer software. Specifically, the agent is evaluated 0.14 and 0.27 points less creative than the software in Scenario A and B, respectively.

Somewhat disappointingly, the results concerning the influence of embodiment on the evaluation of creativity are not statistically significant. Indeed, in both cases the p-value is higher than 0.05, i.e., the value under which the p-value

[4] If the p-value is more than 0.05 there is no strong evidence against the null hypothesis, i.e. the hypothesis that there is no relationship between the variables being considered. From this, however, does not necessarily derive that the alternative hypothesis (i.e. the independent variable does affect the dependent one) is false.

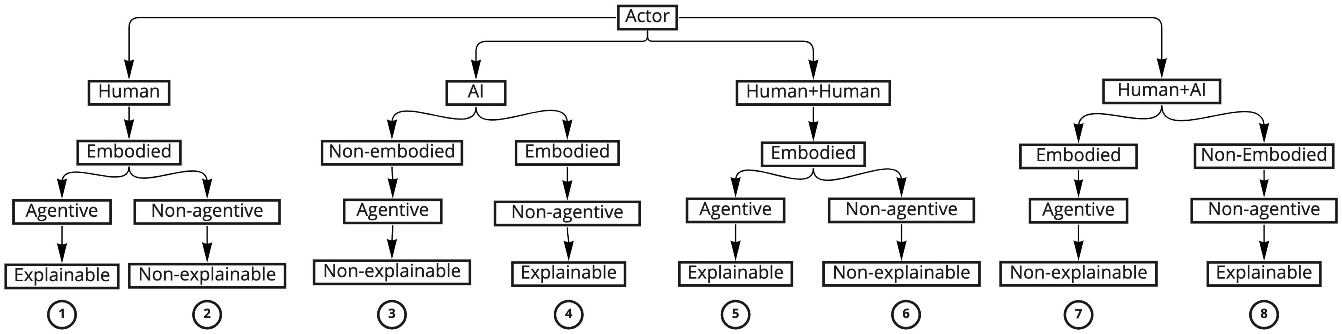


Figure 1:
Vignettes distribution. The scheme shows the distribution of the four dimensions (actor identity, embodiment, agency, and explainability) with their respective levels in the factorial survey experiment. From the distribution resulted 8 distinct vignettes.

	Est.	**Std. err**	**z value**	$Pr(>\|z\|)$
	Painting scenario			
Actor=AI				
Not embod.	0	0	0	0
Embodied	-0.14	0.36	-0.39	0.70
	Vaccine scenario			
Actor=AI				
Not embod.	0	0	0	0
Embodied	-0.27	0.45	-0.61	0.54
	Combined scenarios			
Actor=AI				
Not embod.	0	0	0	0
Embodied	-0.18	0.30	-0.62	0.54

Table 4: *Embodiment Dimension.* The table shows the impact of the manipulation of the Embodiment dimension on the perception of creativity. Baseline is the absence of the attribute.

indicates that the relationship between two variables is statistically significant. Thus, these results give us ground to neither confirm, nor disconfirm the starting hypothesis. Still, while the quantitative analysis of the influence of embodiment on creativity resulting from the factorial survey experiment is not conclusive, more interesting results emerge from the comments to the scenarios left by participants.

Free Response Field

Tables 6 and 7 report some of the participants' comments left in the free response field after completing the factorial survey experiment. Here, respondents were asked to motivate the reasons behind the evaluation of the creativity exhibited by the actors in the scenario they were presented with. When possible, for each vignette (Vig.) are reported comments that are representative of the full range of creativity evaluation scores (Creat.).[5] Keywords are marked in bold by the author, and the categories under which each comment has been collected are indicated in column 2. The list of the categories that emerged from comments is reported in Table 5:

Categories
Anthropomorphism
Autonomy
Collaboration
Data crunching
Problem solving
Randomness
Tool
Training

Table 5: *Categories.* List of categories that emerged from the analysis of the participants' comments through grounded theory methods.

Scenario A (Painting) Considering the painting scenario performed by an individual artificial actor (Vignettes 4 and 3), no meaningful difference emerges between the comments relative to the vignette in which the actor is embodied (Vig. 4) from the ones relative to the vignette in which the actor is a software (Vig. 3). In the comments following a positive evaluation of creativity, 'Autonomy' seems to be the prevailing feature that is attributed to the actor and that, consequently, led to a high rating of creativity (Table 6).

On the other hand, comments following a negative evaluation of creativity, identify the robot or software as a 'Tool' that is, and should, be controlled by human agents. None of the comments relative to these first two vignettes refer to the role that the physicality of the robot might or might not play in performing the action, aside from the action of 'picking colors and tools' that is ascribed to it by participant 1532983358 and that follows a declaration of autonomous decision-making process from the side of the robot itself.

More interesting observations can be made by considering the comments to the vignettes in which human and artificial

[5] Vignette 4 presented an embodied AI (robot), vignette 3 a disembodied AI actor (software), vignette 7 a human and an embodied AI, vignette 8 a human and a disembodied AI (Fig. 1).

Scenario A (Painting)			
Vig.	**Category**	**Creat.**	**Comment**
4	Autonomy	7	The robot was trained and now decides based on the training data. So, it has undergone a process similar to a human learning how to paint. Id. 679187182
	Autonomy	6	The research team did not interfere with the process and Omega decided itself about the process (**picking** colors and tools). Id. 1532983358
	Tool	2	The robot is only an **extension** of the intentions and goals of the human researchers. Id. 1072971333
	Random.	1	The work can be satisfying but cannot count as creative: this is similar to a child who spills paint on a floor in a constellation that looks nice by **accident**. Id. 1727392082
	Tool	0	A robot cannot be creative: it should merely be a **slave for humans**. Id. 1078614007
3	Autonomy	6	The final painting seems to be novel and valuable and produced in a **fairly autonomous way** by Omega. Id. 633966012
	Anthrop.	4	Even though the painting is pleasant, some inner motivation (in the sense of intuition) is missing **bc [sic] it is a software**. Id. 1440542658
	Tool	1	Omega is more like a **tool** rather than an autonomous agent. Omega's agency is limited by the researchers' design goals and intentions. Id. 1072971333
7	Autonomy, Collab.	7	They decided what to do and **acted together**. Id. 178982639
	Collab.	7	**Helen and Omega** created a painting. Together they applied paint on canvas in such a way that they found satisfied their taste and intention. Id. 178982639
	Collab.	6	The **participation of each one of them** and the **interaction between them** is necessary to perform the work. Id. 1702380099
	Collab.	6	There was **collaboration and communication** of some sort between Helen and the robot and I think that is creative. Id. 1206682464
	Tool	4	Helen uses the robot as a **tool**, both for the painting process and for the input for the colour palette. Id. 1724824616
8	Anthrop.	6	Helen clearly has creativity, as for Omega, that would depend upon the underlying architecture. Id. 785499956
	Anthrop.	4	I don't believe we are yet at a stage to give equal ratings to Helen and Omega, the rating is above average **because the human is involved**. Id. 1361386133
	Anthrop.	0	**Software is not creative**. Id. 1150045635

Table 6: *Free responses; Painting scenario.* The table reports some participants' comments in the free responses field after the vignettes based on the painting scenario. Comments have been organized according to creativity score given by respondents and by categories, following a grounded theory method.

actors are collaborating (Vignettes 7 and 8). When the human is presented together with an embodied artificial agent (Vig. 7), the rating of creativity is higher (the lowest rating is 0) and participants explicitly refer to a high level of 'Collaboration' (Collab.) and cooperation that is not indicated in the case of the software and the human as joint actors (Vig. 8). Indeed, when the artificial actor is not embodied, the creativity, when at all recognized, is attributed to the human actor alone (i.e., Helen, see participant 785499956). A categorical refusal at acknowledging the possibility for software to be creative (participant 1150045635) is contrasted in Vig. 7 by a description of the robot Omega as a tool that Helen can use to express her creativity (participant 172482461).

Scenario B (Vaccine) In the comments relative to the scientific discovery scenario, there is no mention of the collaboration that was instead deemed happening between humans and machines in scenario A. The relatively small sample of participants to the study and the low number of significant comments do not allow us to draw a conclusion regarding whether this disparity between the two scenarios is indicative of the fact that human-machine collaboration is deemed more relevant for creativity in artistic than in scientific scenarios. Still, the comments confirm the estimate results obtained from the factorial survey experiment (Table 4) which show that the embodiment of the artificial actor is, slightly, less relevant to creativity in the scientific than in the artistic scenario.

In the vignette presenting the human interacting with the software (Vig. 8) the comments following a positive evaluation of creativity ascribe the latter to the human actor who uses Alpha as a 'Tool' or as a useful, but not autonomous, support (participants 680035971, 2070596251). In the vignette depicting Dr Miller working with robot Alpha (Vig. 7), the robot is recognized as a 'person' by one participant for its creative contribution (participant 1017771618). In general, hesitation at attributing creativity to the artificial actor is observed as coming together with the observation that the action performed is not creative but rather systematic 'Data crunching' (Table 7).

Scenario B (Vaccine)			
Vig.	**Category**	**Creat.**	**Comment**
4	Data crunching	6	There is no bootstrapping by the robot, only exhaustive try-out, computation that is. Still its application worth some creativity. Id. 101174398
	Tool	4	It is using a **tool** (a self-learning machine) to undertake a task. I see this as little more creative than using a supercomputer to break a coded message using brute force. Id. 2006543588
	Data crunching	0	The robot is **systematically** trying all possible combinations of background knowledge, which is the opposite of creatively doing anything. Id. 1100858543
3	Collab.	6	All of this strikes me as hugely creative and **collaborative problem-solving**. Id. 240767967
	Data crunching	5	I don't know if creativity or **computational power** is the better term. Id. 1361386133
	Data crunching	0	I think it is **sophisticated data crunching**, the creativity comes from the initial ideas of the designers. Id. 1724824616
7	Prob. solving	7	A huge amount of creative problem-solving is needed to produce the results described in the story. If a robot is participating creatively, then that robot is, de facto, a **person**, and unambiguously exhibits creativity. Id. 1017771618
	Training	6	They use a lot of **background knowledge and models**, it's a less intuitive process but more logic-based so it's not that creative as the painter. Id. 111980832
	Anthrop.	4	Dr. Miller was indeed creative, but it is difficult to know the role by the robot. Id. 1078614007
8	Tool	7	If we consider Alpha to be a **mathematical structure** (which it is) and if we suppose that Dr. Miller had instead used a different sort of mathematics (and pencil and paper) then we'd not hesitate to ascribe creativity to Miller. By parity of reasoning, this case if creative also. Id. 680035971
	Tool	7	The doctor is utilising Alpha as a **tool, a sophisticated tool** - but in essence no different than a painter's brush. Id. 2070596251
	Tool	4	I think it is not a lot about creativity in this scenario, but about a clever use of a **new (and sophisticated) tool** called Alpha by the scientist. Id. 1440542658
	Data crunching	0	The generation of hypotheses and the evaluation of experiments seems to be things **'canned' algorithms** could do. Id. 2066630687

Table 7: *Free responses; Vaccine Scenario. Free responses; Painting scenario.* The table reports some participants' comments in the free responses field after the vignettes based on the vaccine scenario. Comments have been organized according to creativity score given by respondents and by categories, following a grounded theory method.

Discussion

The reflection on the role of embodiment for the perception of creativity in computational systems is included in a wider discussion on the reception of the engagement of AI systems in the creative sector.

As mentioned, a generalized skepticism against AI engaging in creative activities is well-known and reported by the literature (Moruzzi 2020b; Mumford and Ventura 2015). The acquisition of problem-solving skills, agency, and other features of general intelligence has been indicated as a possible way for AI to gain the appreciation of the public (Bown and McCormack 2009; Gizzi et al. 2020; Moruzzi 2020a; Natale and Henrickson 2022), while other studies report how only the possession of anthropomorphic qualities and a general humanization of technology can lead AI to be perceived as creative (Moruzzi 2020b; Mumford and Ventura 2015; Wyse 2019).

Two obvious limitations of the present study need to be pointed out here: (i) the embodiment dimension comes in degrees, namely the grade of the physical presence of agents and of their interaction with the surrounding environment can vary. In order to conduct a more compelling test on the influence of embodiment on creativity, it would, therefore, be necessary to use more levels in the embodiment dimension and to vary them more accurately. The wider aim of the study that has been presented prevented a more detailed variation of the embodiment dimension. In addition, (ii) in order to obtain more representative and significant results, a bigger and more diverse sample of participants would be necessary. Notwithstanding the value of factorial research methods for assessing the influence of variables on the testing hypothesis, a drawback of this methodology is, indeed, the need for high numbers of participants in order to obtain statistically relevant results for each of the vignettes presented (Auspurg and Hinz 2014). Follow-up research starting from the results of this study will explore the impact that the language used to describe artificial actors has on creativity perceptions (e.g., tool vs collaborator). During the workshop 'The Role of Embodiment in the Perception of Human and Artificial Creativity', as part of ICCC'22, we will expand the methodology followed in the survey presented in this paper, allowing participants to the workshop to assist to live performances by the digital illustrator Renaud Chabrier and by the artist Daniel Berio, who will conduct a procedural generation of graffiti through robotic applications. This 'on-site' study will allow us to obtain more precise and detailed results on the role of embodiment in the judgment of the aesthetic value of an artifact and on the evaluation of the creativity of the process behind its creation.

Conclusions

This paper started with the suggestion made by Guckelsberger et al. (2021) that conducting empirical research on the influence of embodiment on the perception of creativity could contribute to the field of computational creativity as a whole. The paper replied to this suggestion by presenting the results of a recent empirical study on perceptions of creativity in artistic and scientific processes performed by human and artificial agents. The study started with the hypothesis, motivated by previous research, that embodiment positively influences perceptions of creativity. This hypothesis has been tested in the central part of the study through a factorial experiment and the corresponding modulation of the levels of the embodiment dimension. From the results of the evaluation of vignettes in both the artistic and scientific scenario, however, no significant observation could be made. Indeed, the dimension of embodiment had a statistically irrelevant weight on evaluations of creativity.

As partial compensation for the non-conclusive results following the quantitative analysis of the influence of embodiment on creativity, more interesting results have been obtained from the qualitative analysis of the comments left by participants in the free responses section. In particular, what emerges from this study is a higher propensity of respondents in acknowledging collaboration and creative exchange between the human and the artificial actor when the latter is embodied (in the form of a robot). This tendency is observed only in the artistic scenario (Scenario A). What is common to both scenarios, is the description of the artificial actor as a 'Tool' in comments associated to low ratings of creativity.

This indication of the relevance of embodiment in the artistic collaboration between human and artificial actors suggests the importance of exploring the creative potentialities that may emerge from human-machine interaction in the context of artistic processes in further research. The increasingly frequent use of technology in the art sector, indeed, is inevitably bringing with it a modification and development of the relationship between artists, technology, artifacts, and the audience. Importantly, the nature of the human-machine collaboration, as well as the ascription of different characteristics to the machines that may interact with human artists, are dependent on the viewers' perspectives: some may attribute more autonomy to the machine, others may see it as just a tool (Audry and Ippolito 2019).

This, in conclusion, is the ultimate reason for the relevance of the recommendation by Guckelsberger et al.(2021): empirically investigating the perception of creativity and the influence of embodiment on it is crucial for illuminating and suggesting fertile new grounds for co-creativity opportunities. Different kinds of embodiment may generate different modalities of human-machine and machine-machine co-creativity (Davis et al. 2019; Kantosalo and Toivonen 2016; Kantosalo and Takala 2020; Karimi et al. 2018; Saunders and Bown 2015), and this in contrast to the vision of art and artworks as dis-embodied and devoid of any consideration about the context in which they emerge (Audry 2021).

Acknowledgments

I would like to thank the surveyLAB of the University of Konstanz for their support in the data collection and analysis phase. I also thank my colleagues in the Müller AG of the Department of Philosophy, University of Konstanz, for the helpful discussions. This research is part of the projects 'Agency and Scientific Creativity', funded by the Dr. August und Annelies Karst Stiftung and 'The Role of Embodiment in the Perception of Human and Artificial Creativity', funded by the Intersectoral Cooperation Programme, Zukunftskolleg, University of Konstanz.

Author contributions

CM ideated and wrote the paper alone.

References

Audry, S., and Ippolito, J. 2019. Can artificial intelligence make art without artists? ask the viewer. In *Arts*, volume 8, 35. Multidisciplinary Digital Publishing Institute.

Audry, S. 2021. *Art in the Age of Machine Learning*. MIT Press.

Auspurg, K., and Hinz, T. 2014. *Factorial survey experiments*, volume 175. Thousand Oaks, CA: Sage Publications.

Bown, O., and McCormack, J. 2009. Creative agency: A clearer goal for artificial life in the arts. In *European Conference on Artificial Life*, 254–261. Springer.

Brooks, R. A. 1991. New approaches to robotics. *Science* 253(5025):1227–1232.

Byers, G. 2020. Artificial intelligence is restyling the fashion industry.

Charmaz, K. 2006. *Constructing grounded theory: A practical guide through qualitative analysis*. sage.

Cohen, H. 1995. The further exploits of aaron, painter. *Stanford Humanities Review* 4(2):141–158.

Colton, S. 2008. Creativity versus the perception of creativity in computational systems. In *AAAI spring symposium: creative intelligent systems*, volume 8, 7. Palo Alto, CA.

Costello, F. J., and Keane, M. T. 2000. Efficient creativity: Constraint-guided conceptual combination. *Cognitive Science* 24(2):299–349.

Davis, N.; Siddiqui, S.; Karimi, P.; Maher, M. L.; and Grace, K. 2019. Creative sketching partner: A co-creative sketching tool to inspire design creativity. In *ICCC*, 358–359.

Deussen, O.; Lindemeier, T.; Pirk, S.; and Tautzenberger, M. 2012. Feedback-guided stroke placement for a painting machine. Eighth Annual Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging, pp. 25-33.

Foglia, L., and Wilson, R. A. 2013. Embodied cognition. *Wiley Interdisciplinary Reviews: Cognitive Science* 4(3):319–325.

Freedberg, D., and Gallese, V. 2007. Motion, emotion and empathy in esthetic experience. *Trends in cognitive sciences* 11(5):197–203.

Gizzi, E.; Nair, L.; Sinapov, J.; and Chernova, S. 2020. From computational creativity to creative problem solving agents. In *ICCC*, 370–373.

Guckelsberger, C.; Kantosalo, A.; Negrete-Yankelevich, S.; and Takala, T. 2021. Embodiment and computational creativity. *arXiv preprint arXiv:2107.00949*.

Herman, L. M., and Hwang, A. 2022. In the eye of the beholder: A viewer-defined conception of online visual creativity. *New Media Society*.

Huang, C.-Z. A.; Hawthorne, C.; Roberts, A.; Dinculescu, M.; Wexler, J.; Hong, L.; and Howcroft, J. 2019. The bach doodle: Approachable music composition with machine learning at scale. *arXiv preprint arXiv:1907.06637*.

Jean-Pierre, G., and SaId, Z. 2012. The artist robot: A robot drawing like a human artist. In *2012 IEEE International Conference on Industrial Technology*, 486–491. IEEE.

Johnson-Laird, P. N. 1988. Freedom and constraint in creativity. *The nature of creativity: Contemporary psychological perspectives* 202.

Jordanous, A., and Keller, B. 2016. Modelling creativity: Identifying key components through a corpus-based approach. *PloS one* 11(10):e0162959.

Jordanous, A. 2012. *Evaluating computational creativity: a standardised procedure for evaluating creative systems and its application*. Ph.D. Dissertation, University of Sussex.

Jordanous, A. 2016. Four pppperspectives on computational creativity in theory and in practice. *Connection Science* 28(2):194–216.

Kantosalo, A., and Takala, T. 2020. Five c's for human-computer co-creativity-an update on classical creativity perspectives. In *ICCC*, 17–24.

Kantosalo, A., and Toivonen, H. 2016. Modes for creative human-computer collaboration: Alternating and task-divided co-creativity. In *Proceedings of the seventh international conference on computational creativity*, 77–84.

Karimi, P.; Grace, K.; Maher, M. L.; and Davis, N. 2018. Evaluating creativity in computational co-creative systems. *arXiv preprint arXiv:1807.09886*.

Leymarie, F. F.; Bessette, J.; and Smith, G. 2020. *The Machine as Art/The Machine as Artist*. MDPI.

Lubart, T. 2005. How can computers be partners in the creative process: classification and commentary on the special issue. *International Journal of Human-Computer Studies* 63(4-5):365–369.

Marks, J. R. 2015. Isaacson, w.(2014). the innovators: How a group of hackers, geniuses, and geeks created the digital revolution. *Journal of Multidisciplinary Research* 7(1):111–113.

Martin, P. Y., and Turner, B. A. 1986. Grounded theory and organizational research. *The journal of applied behavioral science* 22(2):141–157.

Moruzzi, C. 2020a. Artificial creativity and general intelligence. *Journal of Science and Technology of the Arts* 12(3):84–99.

Moruzzi, C. 2020b. Should human artists fear ai?: A report on the perception of creative ai. In *xCoAx 2020: the Eighth Conference on Computation, Communication, Aesthetics & X*, 170–185.

Moruzzi, C. 2021. Measuring creativity: an account of natural and artificial creativity. *European Journal for Philosophy of Science* 11(1):1–20.

Mumford, M., and Ventura, D. 2015. The man behind the curtain: Overcoming skepticism about creative computing. In *Proceedings of the Sixth International Conference on Computational Creativity June*, 1.

Natale, S., and Henrickson, L. 2022. The lovelace effect: Perceptions of creativity in machines. *New Media & Society* 14614448221077278.

Pease, A.; Colton, S.; Warburton, C.; Nathanail, A.; Preda, I.; Arnold, D.; Winterstein, D.; and Cook, M. 2019. The importance of applying computational creativity to scientific and mathematical domains. In *10th International Conference on Computational Creativity, ICCC 2019*, 250–257.

Pfeifer, R., and Iida, F. 2004. Embodied artificial intelligence: Trends and challenges. In *Embodied artificial intelligence*. Springer. 1–26.

Russell, S. J., and Norvig, P. 2011. *Artificial Intelligence: A Modern Approach. Third Edition*. Upper Saddle River, NJ: Prentice Hall.

Saunders, R., and Bown, O. 2015. Computational social creativity. *Artificial life* 21(3):366–378.

Sharples, M. 1994. Cognitive support and the rhythm of design. In *Artificial intelligence and creativity*. Springer. 385–402.

Simo Linkola, Simo, G. C. M. C., and Kantosalo, A. 2022. How does embodiment affect the human perception of computational creativity? an experimental study framework. *arXiv preprint arXiv::2205.01418*.

Smith, G. W., and Leymarie, F. F. 2017. The machine as artist: An introduction. In *Arts*, volume 6, 5. Multidisciplinary Digital Publishing Institute.

Srikaew, A.; Cambron, M.; Northrup, S.; Peters II, R.; Wilkes, M.; and Kawamura, K. 1998. Humanoid drawing robot. In *IASTED international conference on robotics and manufacturing*. Citeseer.

Tresset, P., and Leymarie, F. F. 2013. Portrait drawing by paul the robot. *Computers & Graphics* 37(5):348–363.

Varela, F. J.; Thompson, E.; and Rosch, E. 2017. *The embodied mind, revised edition: Cognitive science and human experience*. MIT press.

Wyse, L. 2019. Mechanisms of artistic creativity in deep learning neural networks. *arXiv preprint arXiv:1907.00321*.

Yu, D., and Chen, H. 2018. A review of robotic drawing. In *2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, 334–338. IEEE.

From social robots to creative humans and back

Alla Gubenko[1], **Todd Lubart**[2], **Claude Houssemand**[1]
[1]Department of Education and Social Work, University of Luxembourg
2, avenue de l'Université, Esch-sur-Alzette, Luxembourg
[2]Université Paris Cité and Univ. Gustav Eiffel, LaPEA, Boulogne-Billancourt, France
alla@gubenko.eu

Abstract

The research on physically and socially situated artificial agents could complement and enrich computational models of creativity. This paper discusses six perspective lines of inquiry at the intersection of creativity and social robotics. It provides a description of ways in which the field of social robotics may influence (and be influenced by) creativity research in psychology and speculates how human-machine co-creation will affect the notions of both human and artificial creativity. By discussing potential research areas, the authors hope to outline an agenda for future collaboration between creativity scholars in psychology, social robotics, and computer science.

Introduction

The field of Human-Robot Interaction (HRI) provides a fertile environment for interdisciplinary dialogue and reciprocal exchange of results, perspectives, methodologies, and scientific language. This is an ideal context to tackle the problem of human and artificial creativity and study how creative outcomes arise from the interaction between human actors and their social and techno-material environment.

Saunders et al. (2010) and Gemeinboeck and Saunders (2013, 2010) were among the first to recognize the potential of HRI to investigate the enacted and embodied nature of creativity. Beyond the opportunities to interact and improvise with a new kind of creative system, the authors addressed the role of shared physical and social space for the transmission of cultural and tacit human knowledge to robotic agents. Fitzgerald, Goel, and Thomaz (2017) further explored the notions of embodied creativity and human-robot co-creativity in tool-rich human environments and pointed to the challenges and opportunities that physical situatedness of robotic agents poses for computational creativity research. After reviewing recent work related to artistic applications of social robots, Lubart et al. (2021) concluded that, in contrast to 'disembodied' computational models of creativity, physically embodied and socially situated artificial agents, i.e., social robots, afford real-time action and co-creation with humans. The authors argued that social robots represent a potentially efficient ecologically-informed instrument to design, support, and extend human creative thought and action, thus complementing computational creativity research. Figure 1 depicts the *process* of Human-Robot co-creation as the inter-action between human (*person*), robot, and their socio-technical environment (*press*), leading to the emergence of novel and useful *products*.

This article provides an overview of the interplay between social robots and creativity research and outlines possible lines of inquiry at the intersection of these fields. Six perspective research directions are identified: 1) development of methodologies for studying human-robot interaction and co-creation; 2) investigation of human-robot teaming and co-creativity in multiple professional contexts; 3) evaluation of robot's and human-robotic system's creative capabilities and outcomes; 4) development of educational applications of social robots to enhance human creativity; 5) artistic applications of social robots; 6) the use of social robots to emulate the human creative process.

Our intention is twofold. First, we reflect on the current state of research in the field of human-robot interaction and propose possible research directions across disciplinary boundaries. Second, we aim at pointing to the current challenges of existing studies and suggest possible solutions.

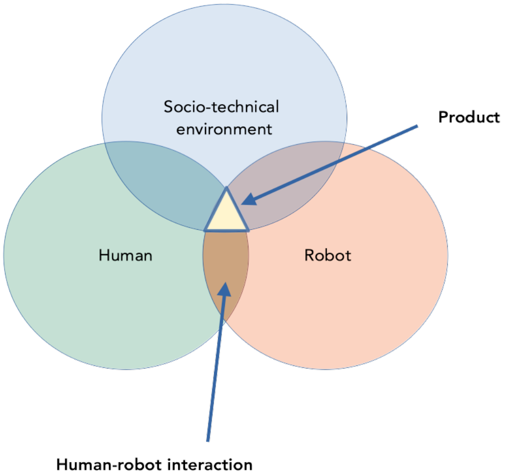


Figure 1: Human-robot co-creation embracing 4 P perspectives on creativity: Person, Process, Press, and Product (Jordanous 2016; Rhodes 1961)

Six lines of research at the intersection of creativity and robotics

A recent review by Guckelsberger et al. (2021) drew attention to the importance of embodiment for computational creativity research and a deeper understanding of human creativity. The authors highlighted the relevance of the 4E Cognition paradigm (Newen, Bruin, and Gallagher 2018; Malinin 2019) for creativity research and called for the embodied computational creativity (CC) research programme. Drawing on this in-depth analysis of embodied CC and recent research in social robotics and cognitive science, below we discuss six perspective lines of inquiry at the intersection of robotics and creativity.

Development of methodologies for studying human-robot interaction and co-creation

As recently noted by Onnasch and Roesler (2021), an increasing variability of existing robots' capabilities and interaction scenarios limits possibilities of comparison and generalization of findings in HRI research. To address this challenge, the authors have proposed a detailed taxonomy to structure and analyse human-robot interaction. Their framework provides three category clusters, such as robot characteristics (e.g., morphology, level of autonomy, task specification), interaction contexts (e.g., field of application, settings), and team classification (team composition, human role, etc). While acknowledging the heuristic value and graphical character of the proposed taxonomy, we suggest that the HRI field may also profit from the adoption of existing methodologies and psychological frameworks to structure different HRI scenarios. Specifically, we see great potential for the application of activity theory initially outlined by Vygotsky (1987) and further developed by Leont'ev (1978), Engeström (1987b), and Kaptelinin and Nardi (2006) as a theoretical lens to formalize the interaction between artificial and human actors.

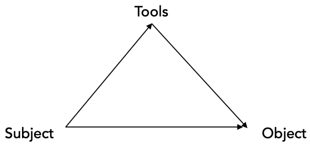


Figure 2: Activity system and two major principles of the activity theory: the tool-mediated character of human activity and its orientation towards an object/outcome.

One of the possible units of analysis in activity theory is an activity system composed of three basic components: subject, tools, and object (outcome) of the activity[1] (fig. 2). Nardi (1996) discussed the resemblance of basic premises of activity theory with theories of situated actions (Suchman 1987) and distributed cognition (Hollan, Hutchins, and Kirsh 2000). Indeed, activity theory is in line with contemporary views of embodied and situated cognition, which consider tools as an organic element of extended cognitive systems (Favela et al. 2021). Engeström (2001; 1987a) also proposed relevant conceptual tools for understanding social action, depicting collaboration as a network of two (or more) interacting activity systems.

Activity level	Question	Description	Example
Activity	Why?	Determined by motives	1a. Building a house; 1b. Completing a software project
Actions	What?	Determined by goals	2a. Laying the foundations; 2b. Programming a module
Operations	How?	Determined by conditions	3a. Using a hammer - grasping, striking; 3b. Using operating system commands

Table 1: Hierarchical structure of activity. Based on Kuutti (1996)

Activity theory considers human behaviour at different levels of abstraction by specifying three possible levels of analysis, ascending from motor operations to complex activities (table 1). Notably, these three levels could be aligned with the three-stratum structure of affordances, proposed under the term *means-end hierarchy* by Vicente and Rasmussen and later elaborated by Wagman, Cialdella, and Stoffregen (2019). Vicente and Rasmussen (1990) suggested that a hierarchically organized set of affordances may be seen as a 'functional landscape' (p.223) through which agents navigate while accomplishing a task.

The concept of affordances has received increased importance in the context of collaborative human-robot activities (Chu and Thomaz 2016; Koppula and Saxena 2015) and creativity research (Kimmel and Hristova 2021; Malinin 2019; Glăveanu 2013). In terms of activity theory, creativity could be re-described as a journey of the actor in interaction with socio-cultural means and tools through a hierarchically organized landscape of affordances towards the production of new and useful artifacts[2].

Advances in the HRI field allow to further develop and adjust activity theory to the current technological context. As such, it could be used as a heuristic model to formalize and understand how human and robotic actors plan their actions and cooperate across three activity levels and multiple interaction layers (Kantosalo et al. 2020) towards a common objective—generating creative artifacts. Different human-robot system configurations could be imagined according to an increased level of robot's autonomy.

[1] For a rigorous and extended description of activity theory and other key components of the activity system: community, rules, and division of labor, see Engeström (1987a; 1987b), and Lindblom and Alenljung (2020), Ceha et al. (2021), Huang and Mutlu (2012) for applications of activity theory in HRI.

[2] In activity theory the artifact is not necessarily material, it could be conceptual or behavioural.

Human-robot teaming and co-creativity in multiple professional contexts

The automation and robotization of human jobs have been considered amongst future global threats, leading to unemployment (Frey and Osborne 2013). Although it is evident that robots will increase their presence in workplace contexts and will automate some routine tasks, in contrast to the 'threatening' view, here we consider possibilities of human-technology teaming (Waefler 2021). In the following, we will speculate on how the role of a robot will depend on how much creativity is needed for the job and how different occupations could benefit from the presence of an embodied artificial agent.

At the first level, we place jobs that eventually necessitate some form of creative problem solving or episodic production of novelty. Examples could be teachers, astronauts, lawyers and alike[3]. At this level, the robot could play a role of a tool in supporting human activity (fig. 3).

An artificial agent might use different strategies to increase human performance and extend the horizon of human action possibilities depending on the stage of the creative process (Amabile 1983; Wallas 1926; Bourgeois-Bougrine et al. 2017):

- Problem definition and representation: suggest searching for alternative formulations of the problem, consider different media to represent it, and look for hidden affordances or relevant problem/object properties and attributes.
- Preparation: find and visualise relevant information or inspiring sets, make mind maps, sketches, planning trees.
- Generation and exploration of possible actions: suggest questioning assumptions, find analogies, use mental/physical synthesis (combination of elements) or disassembly (elimination of elements), search for limitations, potential functions and attributes, means-end analysis, switch attention from problem to the environment, switch mode from generation to action. An artificial agent could also visualise ideas and simulate or model possible movements using their own's bodies.
- Solution evaluation and validation: propose to evaluate the solution from different perspectives, make a SWOT (Strengths, Weaknesses, Opportunities and Threats) analysis, search for alternative actions and strategies, and analyse failures.

As human creativity could be promoted via physical engagement and exploration (Finke, Ward, and Smith 1992; Glăveanu 2012; Suwa, Gero, and Purcell 2000; Schön 1992), robots seem to be a perfect tool that allows humans to alternate and blend thinking and doing. Beyond cognitive support and stimulation, a robot would provide emotional support and stimulate human motivation due to physical presence—an aspect that has become increasingly important during the COVID crisis. One could envisage that a cup of coffee, a hug (or a kiss using the Kissenger machine (Cheok and Zhang 2020)), or verbal encouragement would be beneficial for the creative process. Another useful function of such an agent-for-every-day-problem-solving could be its ability to keep track of human problem-solving efforts and possibility to retain and analyse successful methods and solutions.

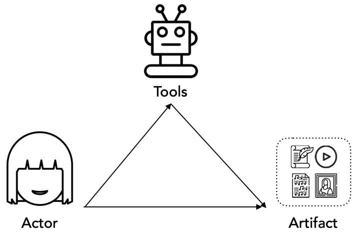


Figure 3: Robot as a tool supporting human creative activity. Adapted from Lubart et al. (2021)

At the second level, we place professions in which the creation of new and valuable artifacts is a necessary part of a job (Koehorst et al. 2019). Professional chefs, art directors, copywriters, and scientists fall into this category. If in the previous scenario, the role of a robot was to inform and stimulate a human actor, this level is marked by an increasing degree of robotic engagement in the human creative process. Beyond the capacities outlined above, a robot is engaged in solution-generation or execution of specific actions and operations set by a human within his or her creative activity. By generating plans and hypotheses and automating human operations, artificial agents would vastly expand the scope and variety of actions available to human actors.

Finally, the third level would be marked by full human-robot teaming, where two activity systems–human and robotic–cooperate in order to achieve a common objective (fig. 4). In the process, they coordinate their activities and synergistically contribute to the production of a novel and valuable artifact. This new type of technologically augmented human creativity (which we call Tech-C) will be paralleled with the emergence of new types of jobs based on mutual inspiration, joint exploration, and co-creation between humans and machines. These new jobs which neither humans nor robots could perform alone should be governed by legal and ethical rules to be developed.

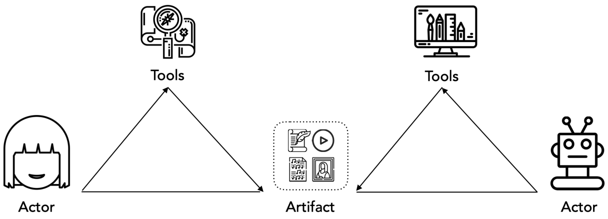


Figure 4: Human-robot co-creation as cooperation of two activity systems. Adapted from Lubart et al. (2021).

[3] We acknowledge that even within these professions the degree of creative intensity could vary and sometimes reach the Pro-C level (Kaufman and Beghetto 2009).

Evaluation of robotic, and human-robotic systems' creative capabilities and outcomes

Increasing human-robotic co-creativity in occupational settings will raise demand for the assessment of creative potentials and evolving creative capacities of robotic and human-robot systems. Developing common metrics to measure robotic capabilities and human-robot interaction is necessary in order to inform future education requirements, anticipate future changes in skill demand (OECD 2021), and improve the performance of human-robot teams (Steinfeld et al. 2006). In this regard, we expect an increasing application of existing human tests and devices as a basis for such assessment.

Not all existing tests would be suitable, however, as many of them are constructed given predominant views of creativity as an essentially 'disembodied phenomenon' that happens mostly in a human mind. Our formalization of creativity as an activity stresses the role of perception and action, as well as symbolic and physical tools for the development of new and useful products. Therefore, below we present examples of possible tests that could be relevant for robotic creativity and human-robot co-creativity assessment accounting for robots' physical embodiment.

- Torrance Thinking Creatively in Action and Movement test (Torrance 1981). This embodied analogue of the Alternative Uses Task (Guilford 1967) would ask a robot to come up and demonstrate multiple ways to do the action (e.g., put a cup in a bin). Initially developed for children starting from 3 years old this test would evaluate the robot's capacity to choose and compose a broad variety of actions to fulfil the same goal. Sufficient behavioural variation along with objects exploration might be key components necessary for innovation and solving new problems in the wild, arguably by increasing the opportunity for learning object affordances and physical properties (Griffin and Guez 2014). This test could be used as inspiration for developing other practical challenges to measure human-robot co-creation.
- Construction using Lego blocks (inspired by Ortiz Jr 2016). A robot would be asked to construct a house using Lego blocks and progressively add new integrated structures such as a garage or a garden. The same task should be completed in multiple possible ways. Collaborative creations could be evaluated via the construction by taking turns with humans.
- Escape-room challenge (inspired by the study by Law, Kasenberg, and Scheutz 2020). A human participant is closed in the room, where the key is out of his/her reach. A social robot capable of moving and understanding human commands is present in the room. The two possible ways to get the key are to use the robot either as a physical tool or a partner to solve the task.

A successful resolution of the proposed challenges involves not only continuous generation of hypotheses and plans but a great extent of exploration of the task's action space. In each case, beyond existing knowledge, the solution depends on the sensorimotor component and the ability to notice and make use of new visuospatial features relevant to the task. These experimental situations testing the actor's behavioral flexibility and ability to improvise solutions with limited available resources have been formalized as MacGyver planning problems (Sarathy and Scheutz 2018) in robotics.

Among other existing tools potentially useful for the evaluation of joint human-robot creation is the Consensual Assessment Technique (Amabile 1982). The technique could be applied for assessing creative artifacts and the gain in the creative output between conditions of human-only creation and creation with a robot[4].

Development of educational applications of social robots to enhance human creativity

An increasing presence of social robots in educational contexts is an established trend (Belpaeme et al. 2018a; Mubin et al. 2013). Studies investigate the educational effectiveness of social robots as tutors (Belpaeme et al. 2018b; Movellan et al. 2009), peers (Kanda et al. 2004; Zaga et al. 2015), and novices (Chase et al. 2009; Tanaka, Cicourel, and Movellan 2007).

In comparison to virtual agents and computer-based tools, physically present systems have numerous advantages when it comes to learning (see Kim and Tscholl 2021, also Li 2015 for review). We also propose that embodied agents support students' situated cognition (Wilson 2002) and learning (Wilson 1993). Situated cognition is coupled with the properties and affordances of settings in which learning takes place and uses these elements to reduce the cognitive workload. Thus, physically present robots have the potential to support such crucial components of scientific discovery as learning by doing, experimentation, observation, and data-driven inferences (see Zimmerman 2007 and Klahr, Fay, and Dunbar 1993 for the description of these components). Active interaction with the environment and hands-on activities, where reasoning and action go in parallel, may allow students to search for evidence not only in the *hypothesis space* of underlying principles but also in the *experiment space* of perceptual cues and patterns. According to Friston et al. (2017) , exploratory behaviour and active sampling of the world often entail unexpected discoveries and may trigger updating learners' explanatory models.

It seems likely that the potential of this technology would expand beyond learning core subjects such as mathematics, reading and science literacy to the development of transversal skills, e.g., critical thinking, creative problem solving, and collaboration. Given the expected increase of robots' participation in occupational fields, early familiarisation with new technology would enable its better acceptance and more fluent and effective human-robot collaboration in the future.

Several recent studies explored the possible benefits of social robots to facilitate creativity in children (Park et al. 2017; Alves-Oliveira et al. 2020; Ali et al. 2021) and adults (Kahn et al. 2016; Alves-Oliveira et al. 2019). In

[4] For further discussion of creativity evaluation in computational co-creative systems see Karimi et al. (2018).

terms of the activity framework, these interventions fall into the application of social robots as a tool to enhance human creative activity. In addition to possible strategies to facilitate the human creative process as outlined in the section devoted to HRI in professional contexts, we expect that social robots could be particularly valuable in the promotion of children's exploration, play, and curiosity, preparing youngsters to adapt to unforeseen circumstances.

Despite its promising potential, this line of research has its pitfalls. Using semi-autonomous or fully tele-operated procedures to enhance creativity with social robots raises the question of whether an eventual effect should be attributed to the robot or human operator. Given this validity issue, interpretation and generalisation of results should be made with caution.

Amplification of artistic applications of social robots

In contrast to the use of social robots as instruments for enhancing the human creative process, researchers started to explore the application of robots as actors participating in creative activity and contributing to the emergence of creative products (Gomez Cubero et al. 2021; Paré 2012; Bretan and Weinberg 2016; Pan, Kim, and Suzuki 2010). We expect that in the next 5 years we will see multiple ways in which the interplay of art and engineering will enrich human artistic culture. Robotic and human actors performing on the theatre stage, human-robot musical bands, and collaborative drawing may open up new forms of art, creating new entry points into robotics and art for children and adults.

Existing examples of making art with robots illustrate moment-to-moment contingency, participatory, and improvisational nature of the creative process. Unfolding through human and robot engagement in shared action, collaborative performance shapes plans and a common vision of the final product. The artistic creative process that arises from human-robot interaction thus represents thus a collaborative dialogic inquiry between participants of the creative process—human artists, machines, materials, and mediating artifacts (Dahlstedt 2012; Ingold 2010). Such physically situated and distributed cognitive systems that coactively exploit and explore affordances and constraints of their surroundings operationalise creative cognition as *creative thinging* (Malafouris 2014), i.e., thinking with and through materials and things.

Human-robot artistic creations integrating and synthetising motion, light, and sound will definitely pose questions of authorship of 'humbot' artifacts. Regardless of whether a social robot could be deemed creative itself and be attributed authorship for its own creation, it is simply a fact that this type of technology will demand humans to be more spontaneous and inventive. Performing with robots which depend on sensory input means that no single linear scenario would be possible. Instead, humans would have to improvise on the fly, imagine multiple alternative paths, and ultimately, develop a larger repertoire of possible actions. This aspect of social robots has the potential to make human-robot cocreation *per se* an ideal training for the unexpected.

Use of social robots to emulate the human creative process

It comes as no surprise that the outlined research directions will be accompanied by continuous efforts to build agents capable to create like humans. Models of the human creative process have been used as inspiration to design creative behaviour in artificial systems (Augello et al. 2016; Hélie and Sun 2010; Vigorito and Barto 2008). Whereas computational models formalize human creativity as a process of solving abstract problems in the absence of a functional body, robots have to deal with the physical world through their sensors and actuators. Although limited by the so-called *curse of dimensionality* (Kober, Bagnell, and Peters 2013, p. 1242), physically and socially present robots afford new and more ecological operationalizations of the creative process and could thus provide additional insight to computational models of creativity.

Guckelsberger et al. (2021) have proposed that robots' sensorimotor capabilities provide an excellent opportunity to examine how creative cognition may be grounded in perception and action. Inspired by recent research in social robotics, Lubart et al. (2021) also suggested that grounding of robots knowledge in vision, audition and proprioception allows to instantiate Ventura's (2016) highest level of computational creativity, where being an embodied author of its sensations a system creates new artifacts based on its own sensorimotor expertise and 'life experience' (see also Colton and Saunders 2018 and Guckelsberger, Salge, and Colton 2017 for further discussion of authenticity, intentionality, and intrinsic motivation in CC systems).

Recently, research has started to address how social robots could demonstrate human-like inventive behaviour in everyday human scenarios, where resources are scarce, and replacement of missing tools is needed (Antunes et al. 2016; Awaad, Kraetzschmar, and Hertzberg 2015). Proposed cognitive architectures allow us to envision social agents capable to improvise solutions for missing equipment by transferring action affordances (Qin, Brawer, and Scassellati 2021; Agostini et al. 2015), discovering new action opportunities (Nyga et al. 2018), and even creating new tools and affordances (Nair and Chernova 2020).

These applications of social robots demonstrate their potential for everyday, little-c creativity (Kaufman and Beghetto 2009), as measured by the Alternative Uses Task. Ironically, as the exact cognitive mechanisms underlying unusual uses are still unknown (but see Gilhooly et al. 2007 and Matheson and Kenett 2020, 2021), robots could help psychologists to unveil the role of language, visual perception, and motor components in performing creative substitutions. The next stage of robots' developmental progression towards creativity would be the development of heuristics permitting agents to choose and evaluate actions based not only on their utility but also on their prospective novelty. One possible way of doing so might be the elaboration of novelty metrics linked to social norms, conventional affordances, and domain standards. Such heuristics estimating 'deviation from normality' and potential utility would enable robots to predict the effect of their action in terms of a potential surprise and value of the final artifact (see Jacob and Magerko 2018 for some examples of possible heuristics).

Conclusion

This paper has attempted to sketch probable future lines of inquiry by crossing interdisciplinary borders of computational creativity, social robotics, and psychology. Imagining and studying possible futures are important to deal better with uncertainties and anticipate opportunities before they emerge and evolve (Broo 2021). We hope that the present work will further stimulate interdisciplinary research investigating the power of embodied agents in relation to the ecological, embedded, enactive, and extended nature of creative cognition.

For centuries, imagination and creativity have been considered as a divine and mysterious spark in humans. Current technological changes allow us to envision a new technologically-augmented type of creativity, in which the inspirational spark would come from the technology and where boundaries between humans and machines would be blurred. We should not forget, however, about the ironies of automation. From one point of view, robotization and increasing human-robot interaction would be the opportunity for humans to offload information and computational processes, freeing up internal capacity for other cognitive and probably more creative tasks (Ecutti, Chemero, and Lee 2021). From a competing point of view, decreasing the frequency of practice of critical creative operations (like idea generation or knowledge retrieval) and outsourcing them to artificial agents could lead to the loss of human creative capacities (Bainbridge 1983). In this regard, the outlined educational interventions, educational robotics (Gubenko et al. 2021), and artistic applications of robots could become critical for preserving human knowledge, flexibility, and the ability to improvise.

Author contributions

AG, TL and CH conceived the article. AG wrote the manuscript and designed figures. TL and CH revised and commented on the text. All authors approved the submitted version.

Acknowledgments

The authors would like to thank three anonymous reviewers for their constructive feedback and thoughtful suggestions on the early version of the manuscript.

References

Agostini, A.; Aein, M. J.; Szedmak, S.; Aksoy, E. E.; Piater, J.; and Worgotter, F. 2015. Using structural bootstrapping for object substitution in robotic executions of human-like manipulation tasks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Ali, S.; Devasia, N.; Park, H. W.; and Breazeal, C. 2021. Social robots as creativity eliciting agents. *Frontiers in Robotics and AI* 8:673730.

Alves-Oliveira, P.; S. Tulli, P. W.; Merhej, R.; Gandum, J.; and Paiva, A. 2019. Sparking creativity with robots: A design perspective. In *Proceedings of the 14th Annual ACM/IEEE International Conference on Human Robot Interaction (HRI)*.

Alves-Oliveira, P.; Arriaga, P.; Cronin, M. A.; and Paiva, A. 2020. Creativity encounters between children and robots. In *in Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*.

Amabile, T. M. 1982. Social psychology of creativity: A consensual assessment technique. *Journal of personality and social psychology* 43(5):997–1013.

Amabile, T. M. 1983. The social psychology of creativity: A componential conceptualization. *Journal of personality and social psychology* 45(2):357–376.

Antunes, A.; Jamone, L.; Saponaro, G.; Bernardino, A.; and Ventura, R. 2016. From human instructions to robot actions: Formulation of goals, affordances and probabilistic planning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 5449–5454.

Augello, A.; I. Infantino, A. L.; Pilato, G.; Rizzo, R.; and Vella, F. 2016. Artwork creation by a cognitive architecture integrating computational creativity and dual process approaches. *Biologically Inspired Cognitive Architectures* 15:74–86.

Awaad, I.; Kraetzschmar, G. K.; and Hertzberg, J. 2015. The role of functional affordances in socializing robots. *International Journal of Social Robotics* 7(4):421–438.

Bainbridge, L. 1983. Ironies of automation. *Automatica* 19(6):775–779.

Belpaeme, T.; Kennedy, J.; Ramachandran, A.; Scassellati, B.; and Tanaka, F. 2018a. Social robots for education: A review. *Science robotics* 3(21):eaat5954.

Belpaeme, T.; Vogt, P.; den Berghe, R. V.; Bergmann, K.; Göksun, T.; Haas, M. D.; Kanero, J.; Kennedy, J.; Küntay, A.; Oudgenoeg-Paz, O.; and Papadopoulos, F. 2018b. Guidelines for designing social robots as second language tutors. *International Journal of Social Robotics* 10(3):325–341.

Bourgeois-Bougrine, S.; Buisine, S.; Vandendriessche, C.; Glaveanu, V.; and Lubart, T. 2017. Engineering students' use of creativity and development tools in conceptual product design: What, when and how? *Thinking Skills and Creativity* 24:104–117.

Bretan, M., and Weinberg, G. 2016. A survey of robotic musicianship. *Communications of the ACM* 59(5):100–109.

Broo, D. G. 2021. Transdisciplinarity and three mindsets for sustainability in the age of cyber-physical systems. *Journal of Industrial Information Integration* 100290.

Ceha, J.; Law, E.; Kulić, D.; Oudeyer, P.-Y.; and Roy, D. 2021. Identifying functions and behaviours of social robots during learning activities: Teachers' perspective. *International Journal of Social Robotics* 1–15.

Chase, C. C.; Chin, D. B.; Oppezzo, M. A.; and Schwartz, D. L. 2009. Teachable agents and the protegé effect: Increasing the effort towards learning. *Journal of Science Education and Technology* 18(4).

Cheok, A. D., and Zhang, E. Y. 2020. Electrical machine for remote kissing and engineering measurement of its remote communication effects, including modified turing test. *Journal of Future Robot Life* 1(1):111–134.

Chu, V.; Fitzgerald, T., and Thomaz, A. L. 2016. Learning object affordances by leveraging the combination of human-guidance and self-exploration. In *11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 221–228. IEEE.

Colton, S., and Saunders, A. P. . R. 2018. Issues of authenticity in autonomously creative systems. In *Proceeding of the 9th ICCC*, 272–279.

Dahlstedt, P. 2012. Between material and ideas: A process-based spatial model of artistic creativity. In McCormack, J., and d'Inverno, M., eds., *Computers and Creativity*. Berlin, Heidelberg: Springer. 205–233.

Ecutti, L.; Chemero, A.; and Lee, S. W. 2021. Technology may change cognition without necessarily harming it. *Nature Human Behaviour* 5(8):973–975.

Engeström, Y. 1987a. Innovative learning in work teams: Analyzing cycles of knowledge creation. In Engeström, Y.; Miettinen, R.; and Punamäki, R., eds., *Perspectives on activity theory*. Cambridge: Cambridge University Pres. 377–404.

Engeström, Y. 1987b. *Learning by Expanding: An Activity-Theoretical Approach to Developmental Research*. Helsinki: Orienta-Konsultit.

Engeström, Y. 2001. Expansive learning at work: toward an activity theoretical reconceptualization. *Journal of education and work* 14(1):133–156.

Favela, L. H.; Amon, M. J.; Lobo, L.; and Chemero, A. 2021. Empirical evidence for extended cognitive systems. *Cognitive Science* 45(11).

Finke, R. A.; Ward, T. B.; and Smith, S. M. 1992. *Creative Cognition: Theory, Research and Applications*. Cambridge: MIT Press.

Fitzgerald, T.; Goel, A. K.; and Thomaz, A. 2017. Human-robot co-creativity: Task transfer on a spectrum of similarity. In *8th ICCC*, 104–111.

Frey, C. B., and Osborne, M. A. 2013. The future of employment: How susceptible are jobs to computerization? *Oxford Martin School*.

Friston, K. J.; Lin, M.; Frith, C. D.; Pezzulo, G.; Hobson, J. A.; and Ondobaka, S. 2017. Active inference, curiosity and insight. *Neural computation* 29(10):2633–2683.

Gemeinboeck, P., and Saunders, R. 2010. Zwischenräume: The machine as voyeur. In *Conf. on Transdisciplinary Imaging at the Intersections between Art, Science and Culture*, 100–109.

Gemeinboeck, P., and Saunders, R. 2013. Creative machine performance: Computational creativity and robotic art. In *ICCC*, 215–219.

Gilhooly, K. J.; Fioratou, E.; Anthony, S. H.; and Wynn, V. 2007. Divergent thinking: Strategies and executive involvement in generating novel uses for familiar objects. *British Journal of Psychology* 98(4):611–625.

Glăveanu, V. P. 2012. What can be done with an egg? creativity, material objects and the theory of affordances. *Journal of Creative Behavior* 46(3):192–208.

Glăveanu, V. P. 2013. Rewriting the language of creativity: The five a's framework. *Review of General Psychology* 17(1):69–81.

Gomez Cubero, C.; Pekarik, M.; Rizzo, V.; and Jochum, E. 2021. The robot is present: Creative approaches for artistic expression with robots. *Frontiers in Robotics and AI* 8:233.

Griffin, A. S., and Guez, D. 2014. Innovation and problem solving: a review of common mechanisms. *Behavioural Processes* 109:121–134.

Gubenko, A.; Kirsch, C.; Smilek, J. N.; Lubart, T.; and Houssemand, C. 2021. Educational robotics and robot creativity: An interdisciplinary dialogue. *Frontiers in Robotics and AI* 8:178.

Guckelsberger, C.; Kantosalo, A.; Negrete-Yankelevich, S.; and Takala, T. 2021. Embodiment and computational creativity. *arXiv preprint arXiv:2107.00949*.

Guckelsberger, C.; Salge, C.; and Colton, S. 2017. Addressing the "why?" in computational creativity: A non- anthropocentric, minimal model of intentional creative agency. In *8th ICCC*, 128–135.

Guilford, J. P. 1967. *The nature of human intelligence*. New York, NY: McGraw-Hill.

Hollan, J.; Hutchins, E.; and Kirsh, D. 2000. Distributed cognition: Toward a new foundation for human-computer interaction research. *ACM Transactions on Computer-Human Interaction (TOCHI)* 7(2):174–196.

Huang, C.-M., and Mutlu, B. 2012. Robot behavior toolkit: Generating effective social behaviors for robots. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction - HRI '12*, 25–32. IEEE.

Hélie, S., and Sun, R. 2010. Incubation, insight and creative problem solving: a unified theory and a connectionist model. *Psychological Review* 117(3):994–1024.

Ingold, T. 2010. The textility of making. *Cambridge Journal of Economics* 34(1):92–102.

Jacob, M., and Magerko, B. 2018. Creative arcs in improvised human-computer embodied performances. In *13th International Conference on the Foundations of Digital Games*, 1–6.

Jordanous, A. 2016. Four pppperspectives on computational creativity in theory and in practice. *Connection Science* 28(2):194–216.

Kahn, P. H.; Kanda, T.; H. Ishiguro, B. T. G.; Shen, S.; Ruckert, J. H.; and Gary, H. E. 2016. Human creativity can be facilitated through interacting with a social robot. In *Proceedings of the 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*.

Kanda, T.; Hirano, T.; Eaton, D.; and Ishiguro, H. 2004. Interactive robots as social partners and peer tutors for children: A field trial. *Human–Computer Interaction* 19(1):61–84.

Kantosalo, A.; Ravikumar, P. T.; Grace, K.; and Takala, T. 2020. Modalities, styles and strategies: An interaction framework for human-computer co-creativity. In *ICCC*, 57–64.

Kaptelinin, V., and Nardi, B. A. 2006. *Acting with technology: Activity theory and interaction design*. Cambridge: MIT press.

Karimi, P.; Grace, K.; Maher, M. L.; and Davis, N. 2018. Evaluating creativity in computational co-creative systems. In *9th ICCC*, 104–111.

Kaufman, J. C., and Beghetto, R. A. 2009. Beyond big and little: The four c model of creativity. *Review of General Psychology* 13(1):1–12.

Kim, Y., and Tscholl, M. 2021. Young children's embodied interactions with a social robot. *Educational Technology Research and Development* 69(4):2059–2081.

Kimmel, M., and Hristova, D. 2021. The micro-genesis of improvisational co-creation. *Creativity Research Journal* 33(4):347–375.

Klahr, D.; Fay, A.; and Dunbar, K. 1993. Heuristics for scientific experimentation: A developmental study. *Cognitive psychology* 25(1):111–146.

Kober, J.; Bagnell, J.; and Peters, J. 2013. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research* 32(1238-1274).

Koehorst, M. M.; van Deursen, A. J.; van Dijk, J. A.; and de Haan J. 2019. Exploring the creative industries: Toward a classification by process and job functions. *Journal of Innovation Management* 7(3):69–95.

Koppula, H. S., and Saxena, A. 2015. Anticipating human activities using object affordances for reactive robotic response. *IEEE transactions on pattern analysis and machine intelligence* 38(1):14–29.

Kuutti, K. 1996. Activity theory as a potential framework for human-computer interaction research. *Context and consciousness: Activity theory and human-computer interaction* 1744.

Law, T.; Kasenberg, D.; and Scheutz, M. 2020. Mate or weight? perceptions of a robot as agent or object in a creative problem solving task. Tufts University, Medford, Human-Robot Interaction laboratory. hrilab.tufts.edu/publications/law2020mate.pdf.

Leont'ev, A. N. 1978. *Activity, Consciousness and Personality*. Englewood Cliffs: Prentice-Hall.

Li, J. 2015. The benefit of being physically present: A survey of experimental works comparing copresent robots, telepresent robots and virtual agents. *International Journal of Human-Computer Studies* (77):23–37.

Lindblom, J., and Alenljung, B. 2020. The anemone: Theoretical foundations for ux evaluation of action and intention recognition in human-robot interaction. *Sensors* 20(15):4284.

Lubart, T.; Esposito, D.; Gubenko, A.; and Houssemand, C. 2021. Creativity in humans, robots, humbots. *Creativity. Theories–Research-Applications* 8(1):23–37.

Malafouris, L. 2014. Creative thinging: The feeling of and for clay. *Pragmatics & Cognition* 22(1):140–158.

Malinin, L. H. 2019. How radical is embodied creativity? implications of 4e approaches for creativity research and teaching. *Frontiers in psychology* 10:2372.

Matheson, H. E., and Kenett, Y. 2020. The role of the motor system in generating creative thoughts. *NeuroImage* 213:116697.

Matheson, H. E., and Kenett, Y. 2021. A novel coding scheme for assessing responses in divergent thinking: An embodied approach. *Psychology of Aesthetics, Creativity, and the Arts* 15(3):412.

Movellan, J.; Eckhardt, M.; Virnes, M.; and Rodriguez, A. 2009. Sociable robot improves toddler vocabulary skills. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction - HRI '09*, 307–308.

Mubin, O.; Stevens, C. J.; Shahid, S.; Mahmud, A. A.; and Dong, J.-J. 2013. A review of the applicability of robots in education. *Journal of Technology in Education and Learning* 1:1–7.

Nair, L., and Chernova, S. 2020. Feature guided search for creative problem solving through tool construction. *Frontiers in Robotics and AI* 7:205.

Nardi, B. A. 1996. Studying context: A comparison of activity theory, situated action models and distributed cognition. In Nardi, B. A., ed., *Context and consciousness: Activity theory and human-computer interaction*. Cambridge: MIT Press. 69–102.

Newen, A.; Bruin, L. D.; and Gallagher, S. 2018. *The Oxford Handbook of 4E Cognition*. Oxford: Oxford University Press.

Nyga, D.; Roy, S.; Paul, R.; Park, D.; Pomarlan, M.; Beetz, M.; and N.Roy. 2018. Grounding robot plans from natural language instructions with incomplete world knowledge. In *in Conference on Robot Learning*, 714–723.

OECD. 2021. *AI and the Future of Skills, Volume 1 : Capabilities and Assessments, Educational Research and Innovation*. Paris: Éditions OCDE.

Onnasch, L., and Roesler, E. 2021. A taxonomy to structure and analyze human–robot interaction. *International Journal of Social Robotics* 13(4):833–849.

Ortiz Jr, C. L. 2016. Why we need a physically embodied turing test and what it might look like. *AI magazine* 37(1):55–62.

Pan, Y.; Kim, M. G.; and Suzuki, K. 2010. A robot musician interacting with a human partner through initiative exchange. In *Proceedings of the 2010 Conference on New Interfaces for Musical Expression (NIME 2010)*, 166–169.

Park, H. W.; Rosenberg-Kima, R.; Rosenberg, M.; Gordon, G.; and Breazeal, C. 2017. Growing growth mindset with a social robot peer. In *Proceedings of the 2017 ACM/IEEE international conference on human-robot interaction*, 137–145.

Paré, Z. 2012. Robot drama research: From identification to synchronization. In *International Conference on Social Robotics*, 308–316. Berlin, Heidelberg: Springer.

Qin, M.; Brawer, J.; and Scassellati, B. 2021. Rapidly learning generalizable and robot-agnostic tool-use skills for a wide range of tasks. *Frontiers in Robotics and AI* 8.

Rhodes, M. 1961. An analysis of creativity. *Phi Delta Kappan* 42(7):305–310.

Sarathy, V., and Scheutz, M. 2018. Macgyver problems: Ai challenges for testing resourcefulness and creativity. *Advances in Cognitive Systems* 6:31–44.

Saunders, R.; Gemeinboeck, P.; Lombard, A.; Bourke, D.; and Kocabali, B. 2010. Curious whispers: An embodied artificial creative system. In *1st ICCC*, 100–109.

Schön, D. A. 1992. *The reflective practitioner: How professionals think in action*. London: Routledge.

Steinfeld, A.; T, F.; Kaber, D.; Lewis, M.; Scholtz, J.; Schultz, A.; and Goodrich, M. 2006. Common metrics for human-robot interaction. In *Proceeding of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction - HRI '06*, 33–40. New York: ACM Press.

Suchman, L. A. 1987. *Plans and situated actions: The problem of human-machine communication*. Cambridge, England: Cambridge University Press.

Suwa, M.; Gero, J.; and Purcell, T. 2000. Unexpected discoveries and s-invention of design requirements: important vehicles for a design process. *Design studies* 21(6):539–567.

Tanaka, F.; Cicourel, A.; and Movellan, J. R. 2007. Socialization between toddlers and robots at an early childhood education center. *Proceedings of the National Academy of Sciences* 104(46):17954–17958.

Torrance, E. P. 1981. *Thinking Creatively in Action and Movement (TCAM)*. Bensenville, IL: Scholastic Testing Service, Inc.

Ventura, D. 2016. Mere generation: Essential barometer or dated concept. In *7th ICCC*, 17–24.

Vicente, K. J., and Rasmussen, J. 1990. The ecology of human-machine systems ii: Mediating direct perception in complex work domains. *Ecological Psychology* 2:207–249.

Vigorito, C. M., and Barto, A. G. 2008. Hierarchical representations of behavior for efficient creative search. In *AAAI Spring Symposium: Creative Intelligent Systems*, 135–141.

Vygotsky, L. S. 1987. Thinking and speech. In Rieber, R., and Carton, A., eds., *The Collected Works of L. S. Vygotsky, Vol. 1. Problems of General Psychology*. New York: Plenum Press. 39–285.

Waefler, T. 2021. Progressive intensity of human-technology teaming. In *in Human Interaction, Emerging Technologies and Future Systems V)*, 28–36. Springer International Publishing.

Wagman, J. B.; Cialdella, V. T.; and Stoffregen, T. A. 2019. Higher order affordances for reaching: Perception and performance. *Quarterly Journal of Experimental Psychology* 72(5):1200–1211.

Wallas, G. 1926. *The Art of Thought*. London: UK: Jonathan Cape.

Wilson, A. L. 1993. The promise of situated cognition. *New directions for adult and continuing education* (57):71–79.

Wilson, M. 2002. Six views of embodied cognition. *Psychonomic bulletin & review* 9(4):625–636.

Zaga, C.; Lohse, M.; Truong, K. P.; and Evers, V. 2015. The effect of a robot's social character on children's task engagement: Peer versus tutor. In *International conference on social robotics*, 704–713. Springer, Cham.

Zimmerman, C. 2007. The development of scientific thinking skills in elementary and middle school. *Developmental review* 27(2):172–223.

Towards Co-Creative Drawing Based on Contrastive Language-Image Models

Francisco Ibarrola
School of Architecture, Design and Planning
The University of Sydney
Sydney, Australia
francisco.ibarrola@sydney.edu.au

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

Kazjon Grace
School of Architecture, Design and Planning
The University of Sydney
Sydney, Australia
kazjon.grace@sydney.edu.au

Abstract

Recent advances in generative machine learning, particularly in the area of text-to-image synthesis, have created huge potential for co-creative systems. It is non-trivial, however, to adapt algorithms intended to generate images that match a given prompt to suit the task of effective collaboration with humans. This paper presents initial experimentation towards developing an agent that can work cooperatively with a human designer in the task of drawing. We do so by utilizing Contrastive Language Image Pretraining (CLIP) to guide the drawing's semantic meaning on a drawing completion process, and fidelity terms to enforce geometric alignment (with what would be the user's in-progress sketch). Preliminary results are presented as a proof of concept, attesting that drawing outputs are both diverse and identifiable as matching the provided prompt, which we interpret as steps towards co-creativity.

Introduction

The traditional conception of the role of a computer within a creative process is that of a tool: precise, effective, and unobtrusive. But today's AI-driven capabilities have started to bend the barrier between tools and collaborators, as reflected on recent studies in Human-Computer Co-Creative Processes (Kantosalo et al. 2020). In this work, we seek to test that barrier further, exploring how generative AI models can be applied to develop co-creative systems that can help designers sketch. There have been many amazing sketching systems developed in the last decade (Davis et al. 2016; Karimi et al. 2018), but several questions remain unanswered before those systems could be applied in practice: Can a co-creative sketching system work towards a user-specified goal? Can it both respect the user's progress and propose modifications when needed? Can it propose small, diverse steps towards completion rather than one-shot autocomplete a drawing? We tackle the task of building a co-creative agent that can answer some of these questions in the affirmative.

For a co-creative drawing agent to be able to be truly cooperative in this context, it should not only be able to pick up on a partial design made by the user, but also to somewhat grasp a sense of its semantic meaning, and produce an output consistent with the user's end goal. Until recently, image generation models were only capable of producing outputs based on simple, specifically trained conditioning labels (Mirza and Osindero 2014). But there has been rapid recent progress in context-agnostic models trained in huge datasets, that have an ability to translate the meaning of complete sentences into images, such as CLIPDraw (Frans, Soros, and Witkowski 2021), and Dall-E (Ramesh et al. 2021).

Our goal in this work is to make progress towards systems with which users can engage in dialogue during creative collaboration, fluidly discussing both strategic and tactical aspects of their emerging design (Bown et al. 2020).

We present a co-creative drawing model that can complete a user's design by taking advantage of CLIPDraw's ability to produce drawings aligned with the semantic meaning of the desired output's description. To this we add loss terms for building on a user's partial sketch and penalising drawing outside a (user-specified) region (see Figure 1).

Related Work

Our co-drawing model builds on CLIPDraw (Frans, Soros, and Witkowski 2021), which built on CLIP (Ramesh et al. 2021), which in turn built on ConVIRT (Zhang et al. 2020).

CLIP and ConVIRT

Contrastive training is based on the following idea: let us consider a batch of (text, image) pairs, $\{(t_n, I_n)\}_{n=1,\dots N}$, paired in the sense that the text t_n describes the image I_n. Then, two functions g and f mapping text and images (respectively) to a latent space $\mathbb{R}^D$ are built using appropriately chosen Neural Network (NN) architectures. These functions are trained to minimize a loss function based on the cosine distance between the image and the text (and vice versa).

$$L_c(t_k, I_k) \doteq -\log \frac{\exp \langle g(t_k), f(I_k)\rangle/\tau}{\sum_{n=1}^{N} \exp \langle g(t_k), f(I_n)\rangle/\tau},$$

where $\langle \cdot, \cdot \rangle$ denotes the cosine similarity and $\tau > 0$ is a scale parameter. Finding g and f minimizing L_c essentially means we are fitting g and f so that t_k is mapped closer to I_k than any other image on the batch. The same is done, using a complementary loss function, to ensure $f(I_k)$ is closer to $g(t_k)$ than to the mapping of any other text within the batch. The result is a shared embedding of both images and text prompts into a space where similarity can be measured between any combination of either. As soon as it was released,

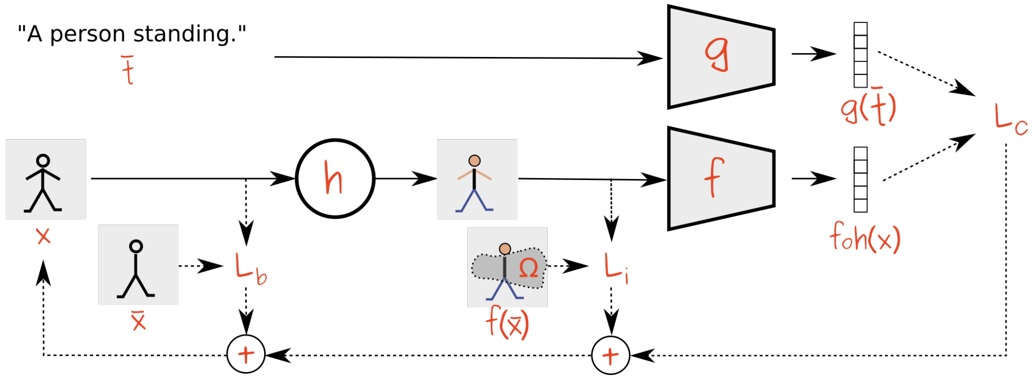


Figure 1: Co-Drawing Model schema: Three different losses are computed on three instances of the pipeline, and the set of Bézier curves x that defines the drawing is optimized with respect to the sum. This ensures parametric similarity with the given curve set $\bar{x}$, consistency with the partial drawing $h(\bar{x})$ and compliance with the semantic meaning $g(\bar{t})$.

CLIP became the focus of a vital and diverse community of online artistic exploration of its capabilities. Much of this exploration was based around generating images from a GAN that match a particular prompt (Liu and Chilton 2021).

CLIPDraw

Of most interest to our goal of co-creative drawing is the recent coupling of the CLIP-based semantic loss (i.e. match to a provided prompt) with a differentiable rasteriser (Li et al. 2020). The resulting system, CLIPDraw, generates an image that fits a provided prompt by manipulating a set of Bézier curves (Frans, Soros, and Witkowski 2021).

Let us denote by $\mathcal{B}$ the space of Bézier curves and let $h : \mathcal{X} \to \mathbb{I}$ be the aforementioned differentiable function that maps the set of finite subsets of $\mathcal{B}$ to the image space. Then, given a set x of Bézier curves, it is possible to build a gradient descent optimization method as:

$$x \leftarrow x + \eta \nabla_x \langle g(t), f \circ h(x) \rangle, \tag{1}$$

where $\eta > 0$ is the learning step.

Put simply, this lets us find a vector drawing that matches a given text prompt, thus enforcing semantic meaning to our model's outputs.

Co-Creative Drawing

Key to co-creative drawing is modifying existing partial sketches. A first instinct upon seeing how CLIPDraw works might be to just let it draw over our partially completed design, since a simple sum over $h(x)$ would preserve the model's differentiability. There are two issues with this approach. Firstly: CLIPDraw can (and often will) simply draw over the partial drawing, completely disregarding the user's design. Secondly, the opposite is also a problem: if the agent is prevented from making any adjustments to the user's input, then it becomes inflexible.

With this in mind, we start by formally defining our partial sketch as a set of Bézier curves $\bar{x} \in \mathcal{X}$, and a text prompt $\bar{t}$ as a string describing the desired end result of our drawing. In practice this partial drawing would be something like an SVG image created by a user.

Curve Fidelity

Let us denote by K_0 the number of Bézier curves in $\bar{x}$ and by K_a the number of additional curves we are going to allow the agent to draw. Finally, let x be the variable associated to the total set of $K = K_0 + K_a$ curves in the model. The idea is that the first K_0 curves produced by the method resemble those of the provided sketch, and we can enforce that by adding the following term to the cost function:

$$L_b(x, \bar{x}) \doteq \sum_{k=1}^{K_0} \sum_{m=1}^{3} \lambda_m \|\bar{x}_k^{(m)} - x_k^{(m)}\|^2, \tag{2}$$

where the index $m = 1, \ldots, 3$ represents one of three variable types: color, coordinates or width, and $\lambda_m > 0$ are regularization parameters, dependant on the type of variable. More specifically, $x_k^{(1)} \in \mathbb{R}^{D_k}$ is a vector containing the path coordinates, $x_k^{(2)} \in [0, 1]^4$ is a vector with the RGBA components of the color of the trace, and $x_k^{(3)} > 0$ represents the width of the trace.

By using this penalisation term, we enforce x to keep the original traces from the partial sketch. Furthermore, by tuning the λ_m parameters, we can control the strength of this constraint, setting large values to strictly maintain the original traces, and smaller values to allow the agent to sensibly move, adjust the width or change the color of the traces.

Drawing within a specified region

Despite the above constraints that enforce similarity on the curves, our agent might still "choose" to draw over the user's partial sketch. To overcome this, we define a region Ω of the canvas where the agent is allowed to draw, by penalizing image discrepancies outside of it. In practice we envisage that this could be provided by the user, or potentially suggested automatically through a process analogous to neural attention.

Notice we want to penalize discrepancies, but not prohibit them. Breaking the rules should always be possible during creative processes, if there is a good reason to do so. To

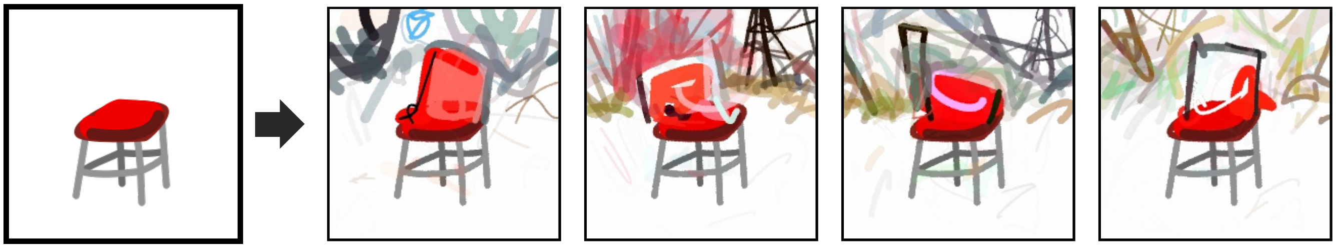


Figure 2: On the left, a user's partial sketch $\bar{x}$. After that, the outputs $h(\hat{x})$ obtained with three random initializations of additional Bézier curves, using the prompt $\bar{t}$ ="A red chair" and the drawing area Ω set as the top half of the canvas.

Algorithm 1 Co-Creative Drawing

```
Set $x_k = \bar{x}_k,\ \forall k = 1, \dots, K_0$.
Let $x_k \sim \mathcal{U}[0, 1],\ \forall k = K_0 + 1, \dots, K$.
Establish a drawing region $\Omega$.
while $\|h(x) - h(x_{(p)})\|_F^2 > \delta$
    $x_{(p)} \leftarrow x$
    $x \leftarrow x - \eta\nabla_x L(x; \bar{t}, \bar{x})$
return $h(x)$
```

accomplish this we define an additional cost function as:

$$L_i(x, \bar{x}) \doteq \alpha\|h(\bar{x}) - h(x)\|^2_{L^2(\Omega^c)}, \tag{3}$$

where $\alpha > 0$ is a regularisation parameter, and $\|\cdot\|_{L^2(\Omega^c)}$ is the L^2 norm defined in the complement of the drawing region Ω.[1] Here again, the fidelity of the image outside the designated drawing area can be enforced (or relaxed) by increasing (or decreasing) the value of α.

Algorithm

Finally, we can add the terms on (2) and (3) to the cosine distance (see Figure 1) to build our overall cost function as

$$L(x; \bar{t}, \bar{x}) \doteq -\langle g(\bar{t}), f \circ h(x)\rangle + L_b(x, \bar{x}) + L_i(x, \bar{x}).$$

The goal is now to find a solution $\hat{x}$ minimizing L. Even though differentiable, L is non-convex and hence finding a global minimum is an intractable problem. Nonetheless, we have found using a gradient descent approach such as (1) often yields good results in practice, and hence we propose to use the method summarised in Algorithm 1.

While the existence of local minima is considered a problem in most settings, it is the opposite here. A high-quality solution $\hat{x}$ within our framework can be understood as one with a low value $L(\hat{x}; \bar{t}, \bar{x})$, while a set of diverse solutions corresponds to a set of elements within different regions of $\mathcal{X}$. This means that the set of highest-quality diverse solutions is a set of local minimizers, and hence a subset of the possible convergence points of the proposed algorithm.

Results

Although creativity is a very tricky concept to define, let alone measure, there is certain consensus on the conjunction of value/utility/quality and novelty/originality/diversity being a good approximation to assess it (McCormack and Gambardella 2022). Both dimensions, however, have their own subjectivity, so we attempt to operationalise them in ways that make sense for co-creative drawing.

As a first intuitive test of our method, we drew a partial sketch, defined a very simple drawing region, ran Algorithm 1 and inspected the outputs (see Figure 2). These images were obtained by providing the agent with a sketch of a stool, and asking it to draw on the top half of the canvas to match the description "A red chair". As a first-order measure of quality: if you the reader were readily able to recognise the drawings as red chairs without reading the caption, then we can attest some subjective standard of quality. Some scribbles appear in the background, which are a consequence of the original CLIP having been trained mostly with natural images, with complexly textured backgrounds. Even ignoring the scribbles, there is also (again, naively) some degree of diversity present among the four chairs, for example in their orientations or the height of their backs.

Quality Assessment

As a simple yet robust way of assessing the quality of the outputs we checked whether CLIP itself recognises the generated drawings as matching the prompt. CLIP can be used as a classifier, with 2343 nouns from its corpus as labels.[2] Evaluating 100 samples from the tasks in Figs 2 and 3 account for a 98% recognition rate for the categories "chair" and "hat", with a confidence of 69.9% ± 19.6%. This accuracy and confidence (estimated as a softmax probability over the very large set of labels) is quite encouraging as a first assessment: our drawings are at least recognisable as the objects they are supposed to be.

Diversity Assessment

Quantifying diversity is yet another task without a standardised method, but recent papers (McCormack and Gambardella 2022) aim to measure it using the intermediate layers of Convolutional Neural Networks (CNNs). It has been shown that different layers encode geometric properties at different scales, which can capture the "style" of images (Ecker, Bethge, and Gatys 2015). Bearing this in mind, we

[1] Better, more complex penalisation functions may be feasible and will be explored in future work.

[2] Ideally, we would want to use a different NN architecture, but to the best of our knowledge, CLIP is the most complete domain-agnostic image classifier currently available.

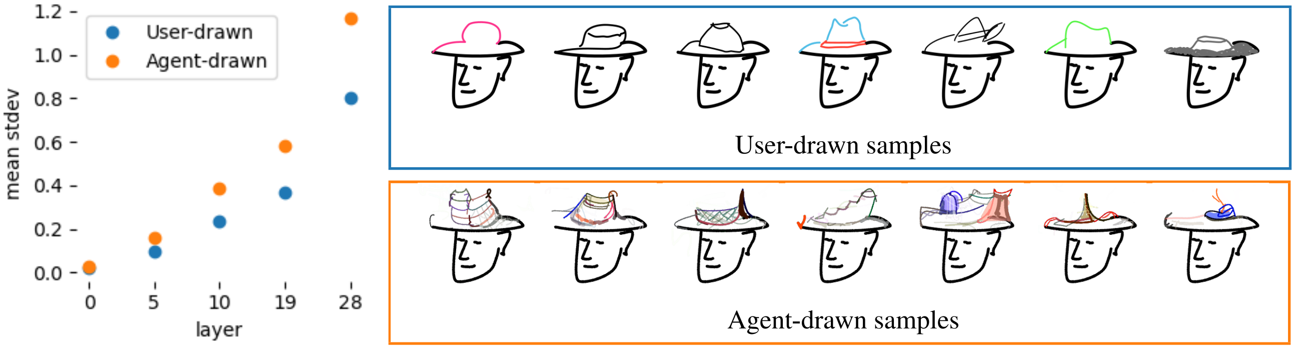


Figure 3: On the left, the mean standard variation over each layer's neuron activations for the 10 tested samples. On the right, some samples of the hat-design task outputs as completed by the users and the agent.

propose to use the variability of the activation of intermediate CNN layers as a measure of diversity.

We provided 10 human subjects with the same partial sketch of a person wearing a hat, and asked them to complete the design as they wish (some samples can be seen in Figure 3). We then put the images through the CNN proposed in (Simonyan and Zisserman 2014) and got the outputs of the five intermediate layers used in Style Transfer. We computed the standard deviation over the 10 samples for every neuron, and averaged over every layer, getting five points of comparison. We then did the same with 10 randomly generated samples from our model. Comparing the two sets (see Figure 3) shows that our generated samples have a higher variance. Although we cannot assure how well these measurements align with our intuitive notion of diversity, the results do suggest at least comparable, if not higher than inter-human design diversity in our results. Of course, this small-scale study has limitations: we neither asked our human subjects to be diverse nor did we recruit skilled milliners to design for us.

Conclusions

We have introduced a model intended for a designer to interact with a sketch-generation agent. Preliminary quantitative results account for the model being capable of producing diverse and quality drawings. Qualitatively, the process and its outputs show potential as a useful fit for co-creative drawing.

The proposed idea is flexible enough to explore the use of other image generative models as the core of the co-creative agent. Future work shall also deal with the formalization and expansion of the introduced experimental setting.

Author Contributions

F. Ibarrola was in charge of defining the cost functions, writing the code, and drafting the paper, and participated in experiment design. O. Bown proposed the initial ideas and did the final reviews. K. Grace was in charge of guidance and editing, and contributed to the model and experiments designs. All the authors participated in the ideas discussions.

Acknowledgments

We would like to acknowledge the Australian Research Council for funding this research (ID #DP200101059).

References

Bown, O.; Grace, K.; Bray, L.; and Ventura, D. 2020. A speculative exploration of the role of dialogue in human-computerco-creation. In *ICCC*, 25–32.

Davis, N.; Hsiao, C.-P.; Yashraj Singh, K.; Li, L.; and Magerko, B. 2016. Empirically studying participatory sense-making in abstract drawing with a co-creative cognitive agent. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*, 196–207.

Ecker, A.; Bethge, A.; and Gatys, L. 2015. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*.

Frans, K.; Soros, L.; and Witkowski, O. 2021. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. *arXiv preprint arXiv:2106.14843*.

Kantosalo, A.; Ravikumar, P. T.; Grace, K.; and Takala, T. 2020. Modalities, Styles and Strategies: an interaction framework for human-computer co-creativity. In *ICCC*, 57–64.

Karimi, P.; Grace, K.; Davis, N.; and Maher, M. L. 2018. Creative sketching apprentice: Supporting conceptual shifts in sketch ideation. In *International Conference on-Design Computing and Cognition*, 721–738. Springer.

Li, T.-M.; Lukáč, M.; Gharbi, M.; and Ragan-Kelley, J. 2020. Differentiable vector graphics rasterization for editing and learning. *ACM Transactions on Graphics* 39(6):1–15.

Liu, V., and Chilton, L. B. 2021. Design guidelines for prompt engineering text-to-image generative models. *arXiv preprint arXiv:2109.06977*.

McCormack, J., and Gambardella, C. C. 2022. Quality-diversity for aesthetic evolution. *arXiv preprint arXiv:2202.01961*.

Mirza, M., and Osindero, S. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; and Sutskever, I. 2021. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, 8821–8831. PMLR.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Zhang, Y.; Jiang, H.; Miura, Y.; Manning, C. D.; and Langlotz, C. P. 2020. Contrastive learning of medical visual representations from paired images and text. *arXiv preprint arXiv:2010.00747*.

D-Graph: AI-Assisted Design Concept Exploration Graph

Shin Sano[1] and Seiji Yamada[2]

[1,2]Department of Informatics, Graduate University for Advanced Studies (SOKENDAI), Hayama, Japan
[1]Institute for Creative Integration, Oakland, CA, United States
[2]Digital Content and Media Sciences Research Division, National Institute of Informatics, Tokyo, Japan
Email: [1]ssano@nii.ac.jp, [2]seiji@nii.ac.jp

Abstract

We present a pilot study of an AI-assisted search tool, the "Design Concept Exploration Graph" ("D-Graph"). It assists industrial designers in creating an original design-concept phrase (DCPs) using a ConceptNet knowledge graph and visualizing them in a 3D graph. A DCP is a combination of two adjectives that conveys product semantics and aesthetics. The retrieval algorithm helps in finding unique words by ruling out overused words on the basis of word frequency from a large text corpus and words that are too similar between the two in a combination using the cosine similarity from ConceptNet Numberbatch word embeddings. Our pilot study with the participants suggested the D-Graph has a potentially positive effect, though we need to improve the UI to help users adhere to the use of the algorithms in the intended ways.

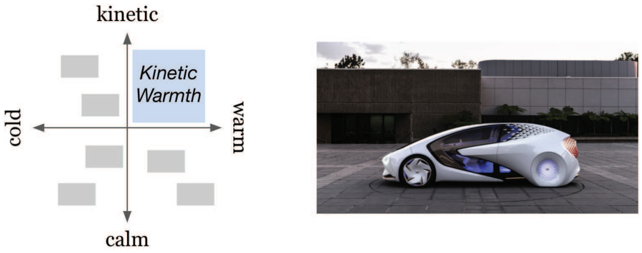


Figure 1: Left: character space for "kinetic warmth." Right: design derived from "kinetic warmth" (© 2021 Toyota Motor Sales, U.S.A., Inc.).

Introduction

We present a pilot study of an AI-assisted search tool, the "Design Concept Exploration Graph" ("D-Graph"). It assists industrial designers in creating an original design-concept phrase (DCPs) using a ConceptNet knowledge graph and visualizing them in a 3D graph. A DCP is a combination of two adjectives that conveys product semantics and aesthetics. The retrieval algorithm helps in finding unique words by ruling out overused words on the basis of word frequency from a large text corpus and words that are too similar between the two in a combination using the cosine similarity from ConceptNet Numberbatch word embeddings. Our pilot study with the participants suggested the D-Graph has a potentially positive effect, though we need to improve the UI to help users adhere to the use of the algorithms in the intended ways.

Designers are in charge of creating the meanings and characters attached to their designs and communicating them with other stakeholders in both visual and verbal modes (Chiu and Shu 2012; Koch et al. 2019; Kita and Rekimoto 2018). We define a design-concept phrase ("DCP") as a combination of two adjectives that conveys product aesthetics. For example, "kinetic warm" was created by the designers at Toyota's North American design studio for Concept-i (Fig. 1-right). This unorthodox DCP was created and communicated using a "character space (CS),"(Fig. 1-left). A character space explains design concepts in terms of how and by which attributes they differ and what already exists or what is to be avoided (Krippendorff 2005). While this approach is common in design practice, there's little computational support for such tasks.

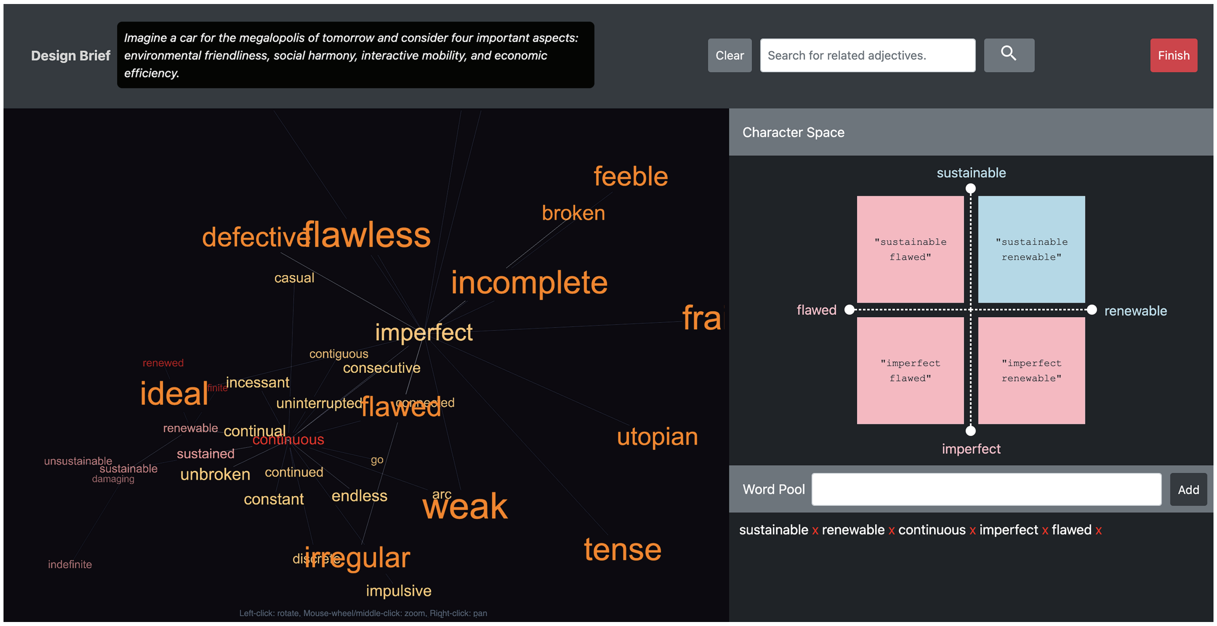
In this study, we focus on two key features: word frequency and cosine similarity between words. Setchi et al.(2011) demonstrated a term with a low document frequency in a corpus could support richer inspiration and creativity for designers. Han et al. (2019; 2018) analyzed the conceptual distances between two ideas expressed in word combinations and concluded that good design concepts have a certain distance between two ideas.

Also, among different language models, the concept distances measured by ConceptNet best agreed with human experts' judgment on concept distance(Han et al. 2020). In D-Graph, we use ConceptNet to measure the cosine similarity of two words. Our method uses it to control the quality of the combinational adjectives that express the design concepts.

Methods

The D-Graph searches for and filters adjectives that are related to users' queries by using a ConceptNet knowledge graph (Speer, Chin, and Havasi 2017).

The top section of the web UI (Fig. 2) has a design brief and a search window. The large space below the design brief is allocated to a "playground," in which graphs of explored words are visualized in 3D hub-and-spoke style. When the user expands the search by clicking words, new clusters are shown in different colors so that users can visually track-



Design Brief

Imagine a car for the megalopolis of tomorrow and consider four important aspects: environmental friendliness, social harmony, interactive mobility, and economic efficiency.

Search for related adjectives.

Word Pool

sustainable x renewable x continuous x imperfect x flawed x

Figure 2: D-Graph web UI (experiment). The baseline tool has Merriam-Webster online thesaurus instead of D-Graph in the playground. All the other UIs are the same for both tools.

back to previous explorations. The lower-right section is a "word pool" where users can store candidate words for design concept phrases. Every time the user puts a query in the search window, clicks on a word on the D-Graph, or drags & drops words from the D-Graph to the word pool, those words are stored in the word pool. Finally, the right-middle section is the CS, which is completed when all four ends on the axes are defined as w_1 through w_4. The words on the CS can be set by dragging & dropping words either from the word pool or directly from the D-Graph. The upper-right quadrant, represented by the combination of w_1 and w_2, is the target design-concept phrase. All the other quadrants are contrasting concepts to be used by the users to explain what are not the target design concepts.

Search and filter algorithms

D-Graph directs users to set words on the CS in a predetermined order (w_1, w_2, w_3, then w_4). This strategy mandates that users first establish the target design concept, represented by the upper-right quadrant (blue square shown in Fig.2) defined by w_1 and w_2, followed by all the other quadrants.

D-Graph has two types of search and filter algorithms, SEARCH_FOR_RELATED_WORDS and SEARCH_FOR_ANTONYMS . The former gets a new word w' from all nodes inside the edge e, using all 33 relations in ConceptNet, except for "Antonym.". Then, each new word w' from the nodes will be filtered in terms of both the relative word frequency ($Freq$) of w' and the cosine similarity ($cosSim$) between the query word w and the new word w', calculated with ConceptNet Numberbatch word embeddings. We currently set the threshold at $(.05 \leq |cosSim| \leq .5)$, according to the results by Han et al. (2020), and $(1 \leq Freq \leq 50)$ from several tests. The latter first gets related words with the former algorithm; then, for each new word w', it gets all the nodes in the edge using the "Antonym" relation. All the results are set as labels of the start node and end node and the link between them to render the graph.

Pilot study design

Ten undergraduate/graduate students (mean age of 25.1 years, $\sigma = 4.01$), in an industrial design department, participated the pilot study. The independent variables were two different search tools, D-Graph (Fig.2), using the above mentioned algorithms, and the baseline tool, using Merriam-Webster online thesaurus instead of D-Graph UI.

The participants were asked to perform the same task twice with the baseline and experimental tools with different design briefs in a counterbalanced order. A design brief is a written description of a project that requires some form of design, containing a project overview, its objectives, tasks, target audience, and expected outcomes (Phillips 2004; Koronis et al. 2018). After reading the brief, the participants were prompted to start the task. First, they were asked to find a combination of two words that forms a DCP by determining w_1 and w_2; then, they were asked to find the opposing concept to each of w_1 and w_2 to generate the CS. The session was concluded when the user was satisfied with the design-concept phrase in terms of w_1 and w_2 and comfortable explaining it in contrast to the other three quadrants. They participated in the experiment online using Playbook UX, a usability testing platform that enables screen recordings. Each participant was given a video instruction and a practice time window (2-3 min.) to get familiar with the tools.

Subjective evaluation A post-task questionnaire with self-reported evaluations was administered using a 7-point Likert scale for four measurements: the “breadth” of exploration that they could perform, the “originality” of the DCP, the “relevancy” of the DCP to the design brief, and the “explainability” of the DCP. The participants were asked to write a short explanation of the DCP (upper-right quadrant of the CS), in contrast to the ideas expressed in the other quadrants. “Explainability” was measured by a 7-point Likert scale on how comfortable they were in explaining the DCP.

Computational metrics The relative word frequency ($Freq$) of both w_1 and w_2 for each DCP as well as the cosine similarity ($cosSim$) between them were calculated post-hoc. The duration of the task and the word count in the “word pool,” which indicates how many words the participant interacted with in the task, were also retrieved. We further analyzed how the selected participants interacted with the words using spatial mapping based on the word embedding.

Qualitative data

All the DCPs and two other words on the CS and the written explanations were obtained. We also had screen recordings that shows the sequence of users’ word explorations.

Results and discussion

All the subjective evaluations on the DCPs with D-Graph were higher than those with the baseline tool, though they were not significant (Table 1). Table 2 shows all the DCPs with the participant ID, the tool used, the mean word frequency ($meanFreq$) of w_1 and w_2, and the cosine similarity ($cosSim$) between them. There were no significant differences ($p = .218$) in mean $cosSim$ between the D-Graph ($.246, \sigma = .195$) and the baseline tool ($.149, \sigma = .124$).

Table 1: Subjective evaluation results

		Ratings (N=10)		
Variable		Bsln. (σ)	Exp. (σ)	p
Breadth	Mean	4.7(1.42)	5.9(1.10)	0.126
	Medien	5	6	
	Mode	6	6	
Originality	Mean	5.1(0.99)	5.4(1.43)	0.591
	Medien	5	6	
	Mode	5	7	
Relevancy	Mean	5.5(1.51)	6.1(0.99)	0.217
	Medien	6	6	
	Mode	7	7	
Explainability	Mean	5.4(1.65)	5.9(1.45)	0.427
	Medien	6	7	
	Mode	7	7	

Table 2: Design concept phrases generated by participants

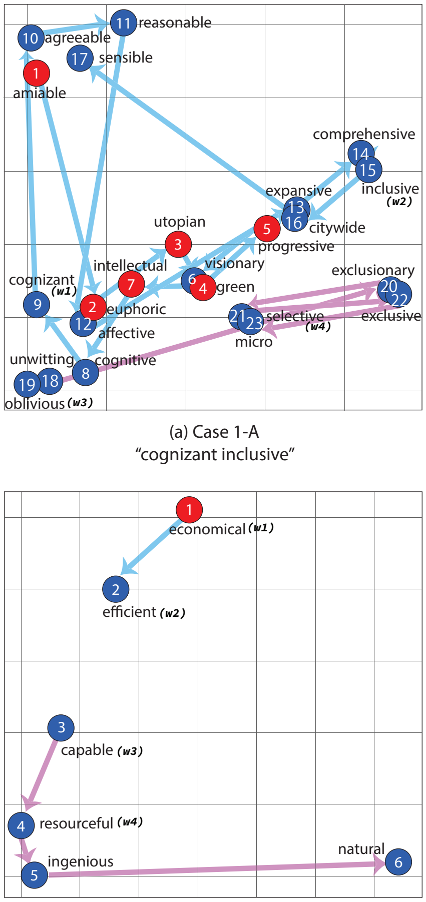
P. ID/Tool	$w_1 + w_2$	$M.Freq$	$cosSim$
1-A/Exp.	“cognizant inclusive”	10.37	0.105
2-A/Exp.	“sustainable renewable”	66.74	0.572
3-A/Exp.	“honest continuous”	26.15	0.123
4-A/Exp.	“futuristic modern”	55.19	0.392
5-A/Exp.	“august renewable”	18.99	0.021
7-B/Exp.	“economical efficient”	31.64	0.551
8-B/Exp.	“affordable neutral”	27.38	0.068
9-B/Exp.	“modular disposable”	5.71	0.162
10-B/Exp.	“empathy transcendent”	1.45	0.240
11-B/Exp.	“utilitarian comfortable”	20.59	0.235
7-A/Bsln.	“efficient functional”	45.45	0.382
8-A/Bsln.	“good-natured safeness”	null	null
9-A/Bsln.	“adventurous lively”	7.01	0.284
10-A/Bsln.	“sustained delightful”	7.41	0.047
11-A/Bsln.	“empathetic minimal”	9.55	0.055
1-B/Bsln.	“protean companionable”	0.13	0.063
2-B/Bsln.	“affordable seamless”	24.26	0.185
3-B/Bsln.	“insensible trustful”	0.18	0.200
4-B/Bsln.	“compact friendly”	28.24	0.121
5-B/Bsln.	“nimble aid”	1.36	0.007

Qualitative results

We will present summaries of two cases in this paper. Fig. 3 shows two cases of the participants’ exploration process. The words are scatter-plotted according to the ConceptNet Numberbatch word embeddings, whose dimensionality is reduced by principal components analysis (PCA).

Case 1-A: “cognizant inclusive” Fig. 3-(a) was created using a D-Graph with design brief A. It had a $cosSim$ value of 0.105 and the $meanFreq$ was (10.37). The number of words in the word pool was 23, and the task duration was 14 minutes and 58 seconds. The words this participant explored aimed to express “being aware of social issues.”. He typed the first word, “amiable,” and used the manual search window instead of clicking the words on the graph until he found the sixth word, “visionary,”. He opened a new tab on the browser and used an online thesaurus to find the adjective form of “utopia” as the system denied it because “utopia” was not an adjective. He also stated, “desirable for sure, but that’s given.” When he stored the 16th word in the word pool, he decided to use “cognizant” and “inclusive” for the DCP. He used “oblivious” for w_3. “Inclusive” on w_2 pulled candidates for w_4, but it showed only four words, including the root node. He tried “micro”, but did not find anything he liked. Therefore, he went back to “inclusive” and tried “exclusive,” which gave him 18 new words. After examining all words there, chose “selective” for w_4. His own ratings for “originality” and “relevancy” were 4 and 7.

Case 7-B: “economical efficient” Fig. 3-(b) was made using a D-Graph with design brief B. It had a $cosSim$ value of 0.551 and the $meanFreq$ was (31.64). The number of words in the word pool was 6, and the task duration was 7 minutes and 1 second. After reading the design brief, this participant typed “economical” in the search window,



(a) Case 1-A
"cognizant inclusive"

(b) Case 7-B
"economical efficient"

Figure 3: Sequence for word exploration in semantic space. Light blue arrows show searches for DCPs for w_1 and w_2, and pink arrows show searches for antonyms for w_3 and w_4. Red circles are users' query inputs in search window. Blue circles are users' clicks on words.

which showed five words. After clicking on "efficient" and "capable," which pulled another 43 words, he spent 1 minute and 40 seconds rotating the graph, moused-over several words to see the definitions, clicked "efficient" and "capable" twice each, and finally cleared the playground and typed "economical" again, followed by clicking "efficient." Then, he clicked "futile," but this was apparently accidental as he deleted "futile" quickly and cleaned up the playground again. He typed and clicked "efficient" and "capable" for the third time. Before clicking the next one, "resourceful," he carefully examined the definitions of "competent," "thorough," and "resourceful." Then, he spent 20 seconds looking at the definition of 'ingenious" and paused another 10 seconds before clicking "ingenious," followed by "natural" for 15 seconds. He further spent 52 seconds rotating the graph, clicked "capable" and "resourceful" again, then put "economical," "efficient," "capable," and "resourceful" for w_1, w_2, w_3, and w_4, respectively. His own ratings for "originality" and "relevancy" were 6 and 7.

Implications for improvement As described above, the participant in case 1-A chose w_2 from the word pool, so he did not utilize SEARCH_FOR_RELATED_WORDS . Yet, he was able to pick two words that were distant enough. He set w_3 and w_4 with words from the D-Graph, which were output according to w_1 and w_2 using SEARCH_FOR_ANTONYMS . This was how we had assumed users would use D-Graph. However, our video analysis unveiled that there were only two cases (4-A and 5-A) that utilized the former algorithm and three cases (1-A, 4-A, and 5-A) that utilized the latter algorithm to explore the words.

For future development, we will add more clarity on what strategy D-Graph helps the users follow. Some participants pointed out the issues in the transparency of the search process and the system status. For example, it was unclear which of the two search algorithms, SEARCH_FOR_RELATED_WORDS or SEARCH_FOR_ANTONYMS, was running. Another option is to implement more automation. For instance, extracting query words from a design brief can be automated. Such automation would lower the initial barrier to exploration.

Different ways of presenting recommended words should also be explored, as it was not easy for some users to avoid cliché words. For example, showing a ranked list of words according to computational linguistic metrics may be an option. In addition, we could further automate the process of concatenating two adjectives in a way that they maintain a certain distance. Finally, we should be investigating engaging factors (Cherry and Latulipe 2014), which we did not measure.

Conclusion

We created an AI-assisted interactive tool, D-Graph, which aims to help industrial designers explore the semantics and aesthetics of design concepts. We integrated two language-based methodologies to attack the problem. 1. We implemented an interactive UI that supports users in broadly exploring words. 2. We implemented search algorithms, utilizing a ConceptNet knowledge graph, that supports users in creating unique compound phrases using an adjective-adjective formula. Our pilot study with 10 student participants did not show significant differences between D-Graph and the baseline tool, which utilizes a conventional online thesaurus. Our qualitative analysis found several important aspects in how users interact with words in lexico-semantic space when searching for words to create a distinguished design concept.

Acknowledgement

We appreciate the support of the Academy of Art University in San Francisco, CA for referring the research participants.

Author contributions

SS contributed to the conception, design, execution of the study, performing the statistical analysis, writing the first draft, and the final version of the manuscript. SY served as the driving force behind the concept, organized the project, and provided guidance throughout the execution of the project. All authors contributed to manuscript revision, read, and approved the submitted version.

References

Cherry, E., and Latulipe, C. 2014. Quantifying the creativity support of digital tools through the creativity support index. *ACM Transactions on Computer-Human Interaction (TOCHI)* 21(4):1–25.

Chiu, I., and Shu, L. H. 2012. Investigating effects of oppositely related semantic stimuli on design concept creativity. *Journal of Engineering Design* 23(4):271–296.

Han, J.; Shi, F.; Park, D.; Chen, L.; Childs, P.; et al. 2018. The conceptual distances between ideas in combinational creativity. In *DS 92: Proceedings of the DESIGN 2018 15th International Design Conference*, 1857–1866.

Han, J.; Park, D.; Shi, F.; Chen, L.; Hua, M.; and Childs, P. R. 2019. Three driven approaches to combinational creativity: Problem-, similarity-and inspiration-driven. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 233(2):373–384.

Han, J.; Hua, M.; Park, D.; Wang, P.; and Childs, P. 2020. Computational conceptual distances in combinational creativity. In *Proceedings of the Design Society: DESIGN Conference*, volume 1, 177–186. Cambridge University Press.

Kita, Y., and Rekimoto, J. 2018. V8 storming: How far should two ideas be? In *Proceedings of the 9th Augmented Human International Conference*, 1–8.

Koch, J.; Lucero, A.; Hegemann, L.; and Oulasvirta, A. 2019. May ai? design ideation with cooperative contextual bandits. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–12.

Koronis, G.; Silva, A.; Kang, J.; et al. 2018. The impact of design briefs on creativity: A study on measuring student designers outcomes. In *DS 92: Proceedings of the DESIGN 2018 15th International Design Conference*, 2461–2472.

Krippendorff, K. 2005. *The semantic turn: A new foundation for design*. crc Press.

Phillips, P. L. 2004. *Creating the Perfect Design Brief: How to manage design for strategic advantage*. Skyhorse Publishing Inc.

Setchi, R.; Tang, Q.; and Stankov, I. 2011. Semantic-based information retrieval in support of concept design. *Advanced Engineering Informatics* 25(2):131–146.

Speer, R.; Chin, J.; and Havasi, C. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. 4444–4451.

Implementation of an Anti-Plagiarism Constraint Model for Sequence Generation Systems

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

Abstract

Sequence generation models are heavily used in computational creative systems in natural language, music composition, and other creative domains. One of the biggest challenges that come with sequence generation models is that, because they learn from existing resources, products from these models often exhibit varying degrees of plagiarism. Papadopoulos, Roy, and Pachet (2014) have, in previous work, presented a max-order Markov automaton to avoid plagiarism in generative sequence models. However, the original publication presented only the algorithmic pseudocode without providing a working implementation. In this replication study, we present a working implementation of the max-order Markov automaton designed to be integrated into sequence generation models for avoiding plagiarism. We use our working implementation to generate new results that verify the efficacy of this approach to avoiding plagiarism. We illustrate how the max-order Markov automaton can be integrated effectively to avoid plagiarism in CC systems using a lyrical music composition system, *Pop**, as an example. Source code:
`https://github.com/aamijani/Anti-Plagiarism_Constraint_Model`

Introduction

Research into the development of generative sequence models have been foundation to much of the advancements in computational creativity (CC) across the domains of music and natural language. As these computationally creative systems gain more attention from the wider population, it becomes crucial for these systems to be aware of and avoid plagiarism (i.e., generation of subsequences longer than some pre-specified length that are copied verbatim from a training corpus). Most of the computationally creative models learn from existing resources to produce new outputs. It is therefore not uncommon for these models to occasionally exhibit plagiarism. Besides the obvious negative impacts that plagiarism can have on the novelty of generative CC systems, the problem of plagiarism also raises an ethical dilemma. The problem of plagiarism in sequence generation models is an imminent problem that must be addressed if CC is to broaden its appeal and relevance beyond being merely an academic pursuit.

Our interest in this study is to replicate the results of *Avoiding Plagiarism in Markov Sequences Generation* (Papadopoulos, Roy, and Pachet 2014). The approach presented in this paper is an effective way to avoid plagiarism. To our knowledge, no one, including the original author, has published an open-source implementation of the model that is available for use. The implementation we present here has been made publicly available. It is implemented using generic type variables allowing for new types to be specified later without need to modify the original codebase. This facilitates integration with Markov generative systems in a variety of domains. A strength of Markov generative systems is that, when combined with constraints (e.g., anti-plagiarism constraints), they are capable of guaranteeing the strict enforcement of those constraints.

Much state-of-the-art sequence generation is currently done both in and out of CC with transformer and LSTM models. For example, ChordAL (Tan 2019) is a system built using Bi-LSTMs that composes melodies. DeepJ (Mao, Shin, and Cottrell 2018) is a generative model that uses LSTMs and is capable of composing music conditioned on a specific mixture of composer styles. GLACNet (Kim et al. 2018) generates visual stories by making use of bidirectional LSTMs. These models have been found to be particularly difficult to constrain. One of the more successful attempts has the Anticipation-RNN model (Hadjeres, Pachet, and Nielsen 2017). However, even this model allows a percentage of generated sequences that do not satisfy constraints and thus still does not make guarantees (Hadjeres and Nielsen 2020).

There have been several Markov generation systems presented in the CC field. For example, *Pop** (Bodily and Ventura 2022) is a music generation Markov model that uses Twitter as an inspiration to produce music. *SMUG* (Scirea et al. 2015) is a system which utilizes Markov chains and works by using academic papers as an inspiration to compose lyrics and melodies. *EMILY* (Shihadeh and Ackerman 2020) is a system that aims to create original poems in the style of renowned poet Emily Dickinson. It makes use of Markov Chains to produce these poems. *LyricJam* (Vechtomova, Sahu, and Kumar 2021) is another generative system that uses live instrumental music to generate lyrics. In order for these and other systems to gain traction beyond merely academic exercises, they need to avoid plagiarism. The success of these and other generative systems depends on their ability to avoid plagiarism.

The study done by Papadopoulos, Roy, and Pachet is important because of how many systems there are that produce music. Moreover, our published model is generalized and is able to not only avoid plagiarism in music generation systems, but also other systems like short-story writing, slogans, etc. In the paper, (2014) introduce a max-order Markov automaton in the framework of constraints satisfaction (CSP). This automaton ensures that sequences generated by a Markov model do not contain subsequences longer than a specified maximum order. Besides its use for avoiding plagiarism, this model can also be used to *detect* plagiarism in existing artifacts (e.g., rhythms, lyrics, etc.).

Replication

The approach outlined by Papadopoulos, Roy, and Pachet (2014) is broken into two algorithms which we refer to as Algorithms 1 and 2. We give a high-level overview of these algorithms below. Our publicly available implementation of these two algorithms can be readily applied to generate sequences of natural language, music, rhythms, etc. In the following sections, we illustrate results of our working implementation in two natural language domains.

Automaton

The base model underlying both a Markov and a max-order Markov model is the finite automaton. A finite automaton $A = \{Q, \Sigma, \delta, q_0, F\}$ is a 5-tuple with elements defined as follows:

- Q is a finite non-empty set of states;
- Σ, the alphabet, is a finite non-empty set of symbols;
- $q_0 \in Q$ is the initial state of the automaton;
- $\delta : Q \times \Sigma \to Q$ is the transition function which maps a state to its successors for a given symbol;
- $F \subseteq Q$ is the set of final or accepting states.

Markov Automaton (Algorithm 1)

The Markov Property states that only the present state (independent of how this state was reached) is the determining factor for the probability of future states. The output of the Markov automaton algorithm is an automaton that recognizes all valid Markovian sequences; i.e., sequences where any two successive N-grams correspond to a $(N+1)$-gram of the training corpus (for a Markov order of N) (Papadopoulos, Roy, and Pachet 2014). A Markov automaton maintains the property that for each $a \in \Sigma$ there exists a unique $q_a \in Q$ and all transitions in δ transitioning via a map to the state q_a.

Fig. 1 shows the 1st-order Markov automaton constructed using 'KALIKIMAKA' as its input dataset. The strength of this (intermediate) model is that it accepts valid Markov strings such as 'MALI' and 'LIMA'. The weakeness of this model is that it also accepts the full original string 'KALIKIMAKA'. For the purposes of eliminating plagiarism, we need to modify the automaton to disallow substrings above a defined length that, albeit valid Markov strings, are also exact substrings of the training set. Thus our Markov automaton is the input to our second algorithm.

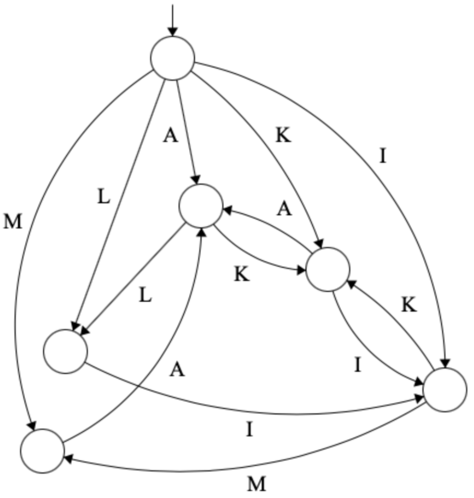


Figure 1: *A Markov automaton for letters*. This automaton accepts all valid Markov strings that can be generated from a 1st-order Markov model trained on 'KALIKIMAKA'. All nodes are accept nodes.

Max-Order Markov Automaton (Algorithm 2)

Algorithm 2 modifies the Markov automaton to remove from the set of accepted strings any sequence containing a 'no-good' subsequence, i.e., a sequence above some length L that appears verbatim in the corpus. This is accomplished by first creating a trie of all no-goods in which all states but the ones corresponding to a full no-good are accept states. This guarantees that a no-good cannot be accepted by the model. Next edges are added for overlapping prefixes. For example, if *ABCD* and *BCEF* are no-goods, then the prefixes *ABC* and *BCEF* share an overlapping prefix (i.e., *BC*). Adding edges for overlapping prefixes ensures that the automaton will not only reject *ABCD* and *BCEF* but also that is will reject *ABCEF*, as well. Algorithm 2 uses an adaptation of the Aho and Corasick (1975) string-matching algorithm to form these cross-prefix transitions.

Fig. 2 shows the resulting max-order Markov automaton derived from the Markov automaton in Fig. 1 with $L = 4$.

Easy reuse in new domains

Our implementation of the max-order Markov automaton uses generics to allow anti-plagiarism constraints to be readily applied to sequence generation models in any domain. Whereas our previous examples demonstrated the construction a max-order Markov automaton for constraining sequences of *letters*, we demonstrate here the application of our implemented model to constrain sequences of *words*. Fig. 3 shows the Markov automaton derived from the training sequence 'can you can a can as a canner can can a can'. A expected, the model accepts valid Markov strings such as

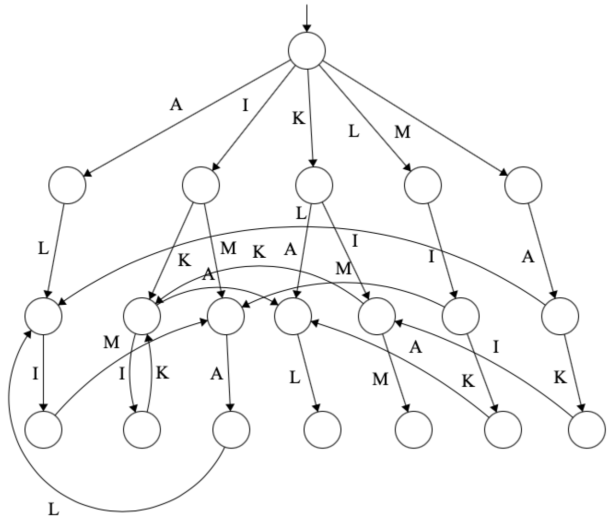


Figure 2: *A max-order Markov automaton for letters*. This automaton accepts the same set of strings as the automaton in Fig. 1 minus strings of length ≥ 4 that contain exact substrings of (i.e., plagiarize) the training sequence 'KALIKIMAKA'. All nodes are accept nodes.

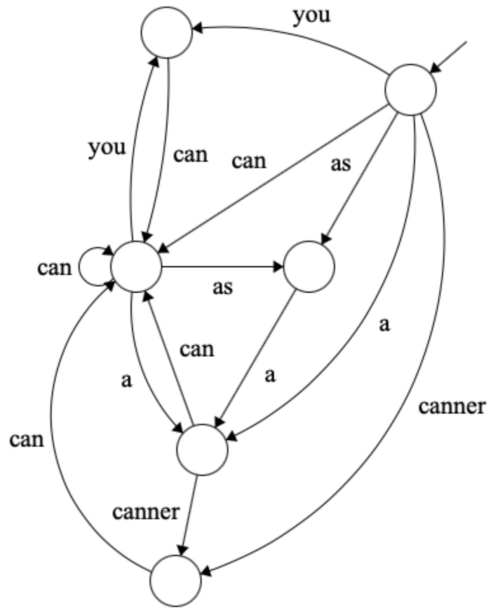


Figure 3: *A Markov automaton for words*. This automaton accepts all valid Markov sequences that can be generated from a 1st-order Markov model trained on 'can you can a can as a canner can can a can'. All nodes are accept nodes.

'you can can a canner' and 'can a canner can you' as well as the full original sequence.

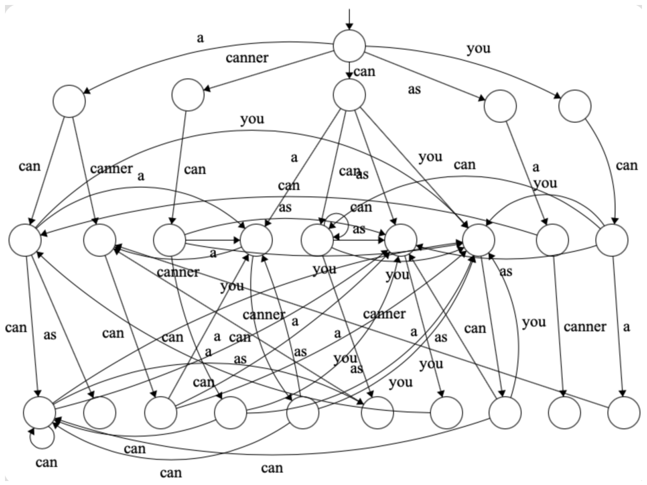


Figure 4: *A max-order Markov automaton for words*. This automaton accepts the same set of sentences as the automaton in Fig. 3 minus sentences of ≥ 4 words that contain exact phrases from (i.e., plagiarize) the training sequence 'can you can a can as a canner can can a can'. All nodes are accept nodes.

Integrated Visualization Feature

Our implementation of the algorithms for constructing max-order Markov automata includes a feature allowing graphs of the finite automata to be visualized at each intermediate algorithmic step and/or in their final form. This enables users to better see and understand the process of how the automata are built and to verify the model's results. The feature saves graphs in .dot format.

Applications in CC Systems

Our primary motivation for being able to generate max-order Markov automata is to incorporate anti-plagiarism constraints into *Pop**, a CC lyrical music composition system built using constrained Markov models (Bodily and Ventura 2022). The model uses Markov models to generate interdependent harmonic, melodic, rhythmic, and lyrical sequences. Like several other generative models (cf. (Fargier and Vilarem 2004; Papadopoulos et al. 2015)), *Pop** defines and integrates constraints in the form of finite automata. For example, Fig. 5) illustrates a finite automaton constructed to enforce a rhyming constraint between the first and fourth words in a four-word lyrical sequence. Automata such as these are then compiled with Markov models to probabilistically generate sequences that adhere to constraints.

Computational theory informs us that regular languages are closed under intersection, and indeed algorithms have been presented that, given two automata A and B, create a third automata C, such that the set of sequences accepted by C is the intersection of the sets accepted by A and B (Sipser 1996). By combining max-order Markov automata with the automata already in use to constrain the generation of *Pop**, we immediately inherit the ability to constrain our compositions against plagiarism—across all aspects of the

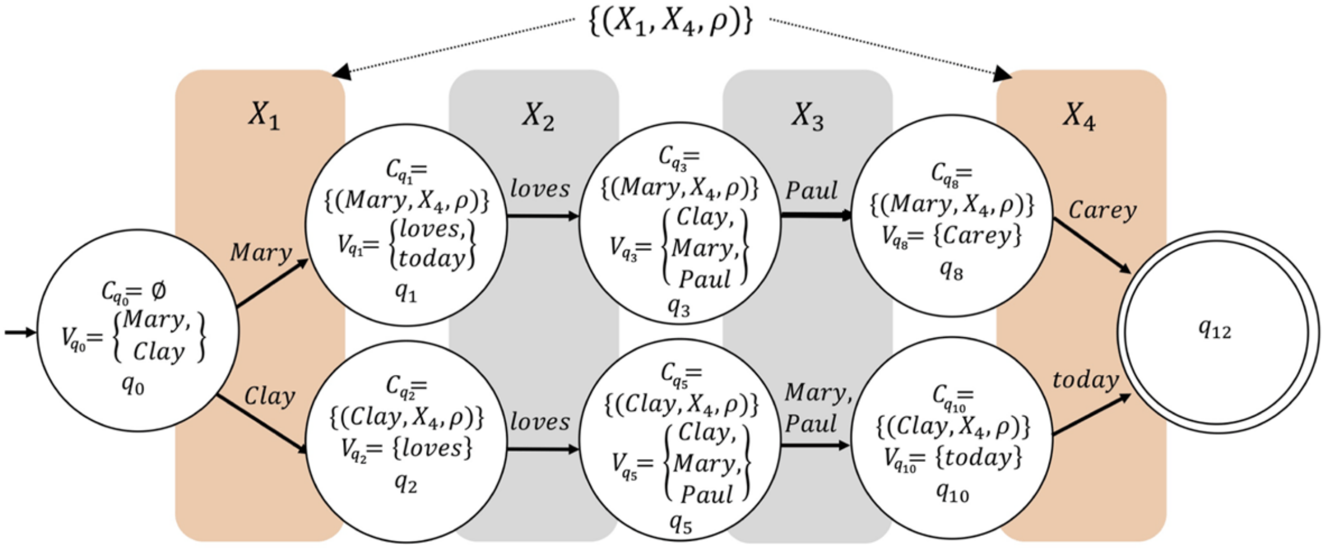

$\{(X_1, X_4, \rho)\}$
X_1
X_2
X_3
X_4
$C_{q_0} = \emptyset$
$V_{q_0} = \{Mary, Clay\}$
q_0
Mary
Clay
$C_{q_1} = \{(Mary, X_4, \rho)\}$
$V_{q_1} = \{loves, today\}$
q_1
$C_{q_2} = \{(Clay, X_4, \rho)\}$
$V_{q_2} = \{loves\}$
q_2
loves
loves
$C_{q_3} = \{(Mary, X_4, \rho)\}$
$V_{q_3} = \{Clay, Mary, Paul\}$
q_3
$C_{q_5} = \{(Clay, X_4, \rho)\}$
$V_{q_5} = \{Clay, Mary, Paul\}$
q_5
Paul
Mary, Paul
$C_{q_8} = \{(Mary, X_4, \rho)\}$
$V_{q_8} = \{Carey\}$
q_8
$C_{q_{10}} = \{(Clay, X_4, \rho)\}$
$V_{q_{10}} = \{today\}$
q_{10}
Carey
today
q_{12}

Figure 5: Shown is an automaton designed to enforce a rhyming constraint, ρ, between the first and last positions, X_1 and X_4, in the Markov generation of a four-word sentence in the CC music composition system *Pop**. Generative systems like *Pop** that define constraints using automata are particularly well-suited for easy integration of max-order Markov automata for constraining against plagiarism. Figure originally from (Bodily and Ventura 2022).

composition.

Conclusion

We have presented an implementation of an anti-plagiarism model first presented by Papadopoulos, Roy, and Pachet (2014). The model works by utilizing a max-order Markov automaton that only accepts non-plagiaristic sequences based on a specified corpus. We illustrated through examples how, through the use of generics, this model can be applied with constrained sequence generation in novel CC domains, and highlighted, in particular, its envisioned integration into a lyrical music composition system. Whether the goal be to achieve greater novelty or to show increased respect to the ethics of avoiding plagiarism, the implemented model we have presented will serve to aid CC practitioners to achieve greater and more ambitious milestones in the pursuit of computational creativity.

Author Contributions

Both authors contributed to all aspects of the work, including ideation, narrative/position development and writing.

Acknowledgments

The authors have no acknowledgments.

References

Aho, A. V., and Corasick, M. J. 1975. Efficient string matching: an aid to bibliographic search. *Communications of the ACM* 18(6):333–340.

Bodily, P., and Ventura, D. 2022. Steerable music generation which satisfies long-range dependency constraints. *Transactions of the International Society for Music Information Retrieval* 5(1).

Fargier, H., and Vilarem, M.-C. 2004. Compiling csps into tree-driven automata for interactive solving. *Constraints* 9(4):263–287.

Hadjeres, G., and Nielsen, F. 2020. Anticipation-rnn: Enforcing unary constraints in sequence generation, with application to interactive music generation. *Neural Computing and Applications* 32(4):995–1005.

Hadjeres, G.; Pachet, F.; and Nielsen, F. 2017. Deepbach: a steerable model for bach chorales generation. In *International Conference on Machine Learning*, 1362–1371. PMLR.

Kim, T.; Heo, M.; Son, S.; Park, K.; and Zhang, B. 2018. GLAC net: Glocal attention cascading networks for multi-image cued story generation. *CoRR* abs/1805.10973.

Mao, H. H.; Shin, T.; and Cottrell, G. 2018. Deepj: Style-specific music generation. In *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*, 377–382.

Papadopoulos, A.; Pachet, F.; Roy, P.; and Sakellariou, J. 2015. Exact sampling for regular and markov constraints with belief propagation. In *International Conference on Principles and Practice of Constraint Programming*, 341–350. Springer.

Papadopoulos, A.; Roy, P.; and Pachet, F. 2014. Avoiding plagiarism in markov sequence generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.

Scirea, M.; Barros, G. A.; Shaker, N.; and Togelius, J. 2015. Smug: Scientific music generator. In *ICCC*, 204–211.

Shihadeh, J., and Ackerman, M. 2020. Emily: An emily dickinson machine. In *ICCC*, 243–246.

Sipser, M. 1996. Introduction to the theory of computation. *ACM Sigact News* 27(1):27–29.

Tan, H. H. 2019. Chordal: A chord-based approach for music generation using bi-lstms. In *ICCC*, 364–365.

Vechtomova, O.; Sahu, G.; and Kumar, D. 2021. Lyricjam: A system for generating lyrics for live instrumental music. *arXiv preprint arXiv:2106.01960*.

3. Theory and problem solving

Open Computational Creativity Problems in Computational Theory

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

Despite clear benefits that would derive from their development, applications of computational creativity (CC) in math, science, and logic are heavily underrepresented in comparison with more artistic domains. In this paper, we examine the application of CC in the domain of computational complexity theory and identify several problems in the domain to which CC might be applied. In particular, we propose and define the task of creating reductions between NP-complete problems, the (sub)task of creating gadgets for use in constructing such reductions, and the task of gamification of reductions and argue that each of these may be addressed as interesting, fruitful CC challenge problems.

Introduction

Arguably the greatest achievements in human creativity have been in the fields of science, mathematics, and technology. And yet a 2017 review of application domains considered in computational creativity (CC) found that only 3% of 353 papers published over the preceding 12 years fell under the category of "Math, Science, and Logic" (Loughran and O'Neill 2017). This gap has been frequently mentioned in CC literature, and efforts have repeatedly been made to highlight the importance of applying CC in scientific and mathematical domains (Pease et al. 2019).

Computational complexity theory (CCT) represents a subfield of theoretical computer science that focuses on the classification of problems based on resource usage (e.g., time and memory) as well as how problems within and between complexity classes relate to one another. The classification of problems according to their complexity has profound real-world implications for the types of solutions that should be pursued for a particular problem and whether such solutions can be expected to be optimal. Besides providing mechanisms for proving the complexity of a particular problem, CCT also provides tools that can facilitate the reuse of existing algorithms to solve new problems.

In this paper we focus on the particular CCT subtopic of NP-completeness. Contributions in this subdomain tend to be impactful because such problems are ubiquitous in the real world and lie just beyond the grasp of modern computers when it comes to finding optimal solutions. NP-complete problems tend to take the form of optimization or decision problems with even minor improvements in algorithmic performance leading to significant cost savings in terms of time, energy, money, accuracy, or other value metrics. For this reason NP-complete problems have been studied in areas as diverse as advanced manufacturing (e.g., optimization of production lines, route inspection); computing/data/visualization (e.g., modularity maximization for graph visualization); homeland and cybersecurity (e.g., assembling an optimal Bitcoin block, cryptography); energy policy (e.g., the graph bandwidth problem in electronic design automation); energy-water (e.g., optimizing power/water flow across a network); innovative energy systems (e.g., the formulated energy and content aware vessel throughput maximization problem); and nuclear energy (e.g., the berth allocation problem as applied to reloading nuclear core fuel assemblies). The list of NP-complete problems grows ever longer.

We consider the main goal in this paper to be the framing and formal articulation of four important open problems in computational theory as CC problems. These problems are defined by characteristics that are typical of CC problems: each requires generation of a creative solution in a context with relatively well-established definitions of typicality, novelty, intention, and value. Similar to creating mathematical proofs, these problems are generally difficult even for trained humans. However, just like mathematical proofs, there are strategies that humans use that can aid in articulating a structured, generative process. Prerequisite to making substantive progress attempting solutions to these problems, the CC field needs a precise definition of these problems together with a clear understanding of the evaluative criteria associated with each. In essence, we aim to open a new potential subdomain of computational creativity to the CC field—the domain of NP-completeness in CCT—or, in other words, to bring awareness of the potential impact that computational creativity could have in a domain that has hitherto not been considered in the field of CC. We aim not merely to introduce the subdomain, but to articulate problems within this domain well enough that CC researchers will immediately be able to begin to innovate and implement CC solutions to these problems. Though the domain deals with theory, the practical implications are immediate and significant, and we will seek to highlight these as well.

Computational Complexity Theory

While there are many approaches to treating computation, we will find it convenient to consider computation from the perspective of determining set membership as, for example, is done in (Sipser 2013). Given a set of symbols (alphabet) Σ, we can define the set of all strings over that alphabet as Σ^*. We can then define a *language* $A \subseteq \Sigma^*$ as a set of strings. A language A is *decidable* if there exists a computable function $f_A : \Sigma^* \to \{0,1\}$ such that[1]

$$f_A(w) = \begin{cases} 0 & \forall w \notin A \\ 1 & \forall w \in A \end{cases}$$

and we say that the language of f_A is A, $L(f_A) = A$. We can define computation as the problem of determining whether some string $w \in \Sigma^*$ is a member of a particular language A. For this reason, we use the terms *language*, *decision problem* (or simply *problem*), and *set* interchangeably. When speaking of decision problems, a string w being considered for membership in a set A is called an *instance* of problem A. As a running example, we will look at two decision problems in particular: 3SAT and CLIQUE.

3SAT In logic and computer science, a *Boolean literal* is either a variable, called a *positive literal*, or the negation of a variable, called a *negative literal*. A *clause* is a disjunction of literals (or a single literal). A *Boolean formula* is in *conjunctive normal form* (CNF) if it is a conjunction of clauses (or a single clause). A formula ϕ is 3CNF if the formula is in CNF and each clause in ϕ contains exactly 3 literals. Given a 3CNF Boolean formula ϕ, the 3SAT problem is to determine whether ϕ is satisfiable, i.e., whether or not there exists an assignment to each variable in ϕ such that ϕ evaluates to true. Using $\langle\phi\rangle$ to denote the string representation of ϕ, 3SAT is defined as the following decision problem:

$$\text{3SAT} = \{\langle\phi\rangle \mid \phi \text{ is a satisfiable 3CNF formula}\} \tag{1}$$

A specific example of an instance of 3SAT is shown in Figure 2a. Many real-world problems in domains such as artificial intelligence, circuit design, and automatic theorem proving are representable as 3SAT instances.

CLIQUE In graph theory, a *graph* $G = (V, E)$ consists of a set $V = \{v_0, \ldots, v_n\}$ of *nodes* or *vertices* and a set of edges E. For *directed graphs* an edge $e = (v_i, v_j)$ is an ordered pair where order indicates the direction of the edge; for *undirected graphs* an edge $e = \{v_i, v_j\}$ is an unordered pair. A *clique* in an undirected graph is defined as a subset of nodes $V' \subseteq V$ for which $\forall v_i, v_j \in V', \{v_i, v_j\} \in E$. Given a graph G and an integer k, the CLIQUE problem is that of determining whether or not there exists a clique in G of size $\geq k$. Using $\langle G, k\rangle$ to denote the string representation of a G, k pair, CLIQUE is defined as the following decision problem:

$$\text{CLIQUE} = \{\langle G, k\rangle \mid G \text{ contains a clique of size } \geq k\} \tag{2}$$

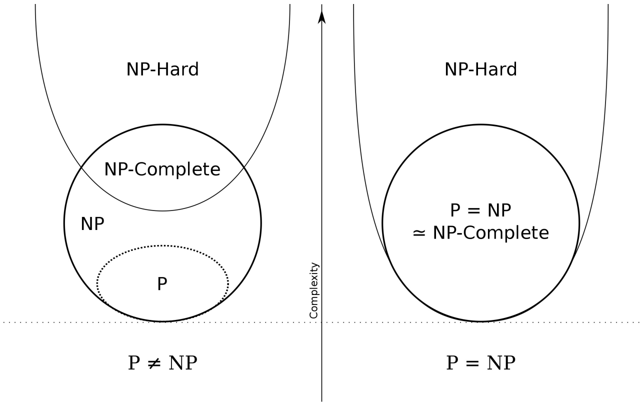


Figure 1: *Does P = NP?*. Two different views of some important computational complexity classes. It is unknown which view is correct, though most researchers believe P $\neq$ NP. If this is the case, several important theoretical questions about the class NP-complete, with significant, practical implications, provide interesting potential for CC research.

An instance of the CLIQUE problem is shown in Figure 2b. The CLIQUE problem has been used to represent instances of many real-world problems in domains such as social networks, bioinformatics, and computational chemistry.

The Theory of NP-completeness

The theory of NP-completeness offers a way to classify decision problems according to their inherent complexity. A problem is said to be in the class P if it can be decided (i.e., solved) by a polynomial-time algorithm.[2] A problem is said to be in the class NP if a solution to the problem (sometimes called a *certificate*) can be verified to be correct by a polynomial-time algorithm. Clearly problems that can be solved in polynomial time can also be verified in polynomial time, and therefore P $\subseteq$ NP. It is an open question of broad interest whether P $=$ NP or P $\neq$ NP (see Figure 1).

The fascination with these two particular complexity classes stems from the fact that only polynomial-time algorithms can be effectively computed in reasonable time by classical computers for non-trivially-sized inputs. For all practical purposes, most computer scientists assume P $\subset$ NP, and this belief is largely perpetuated by the existence of a third class, NPC, of problems called *NP-complete* problems. This is a unique class of NP problems that are stubbornly resistant to being solvable by polynomial-time algorithms, and yet no one has been able to prove this barrier actually exists. NP-complete problems are considered the hardest problems in the class NP. But what makes them most fascinating is that every NP-complete problem is a *gateway problem*: the existence of a polynomial algorithm for deciding any one of them would mean that the entire class of languages is decidable in polynomial time. To be more specific, every NP-complete problem A can be reduced to every other NP-complete problem B (written $A \leq_{\text{P}} B$)

[1] e.g., a Turing machine that halts with 0 on its tape $\forall w \notin A$ and halts with 1 on its tape $\forall w \in A$.

[2] A polynomial-time algorithm is an algorithm whose run time can be bounded with a polynomial function of the size of the input.

via some polynomial-time reduction algorithm. As a consequence, if one NP-complete problem B is one day discovered to have a polynomial-time solution algorithm, then by transitivity every other NP-complete problem A can be solved in polynomial-time by first reducing it in polynomial-time to B and then using the polynomial-time solver of B to find a solution to A. This is the basis for proofs of NP-completeness—a language B is NP-complete if it can be shown that:

1. $B \in \text{NP}$
2. $\forall A \in \text{NP}, A \leq_{\text{P}} B$

NP-complete problems are ubiquitous in the real-world and play prominent roles across nearly every field,[3] and it is common for those tasked with solving such problems to use this approach to prove NP-completeness in order to justify the use of heuristic or approximation algorithms when solving such problems. Requirement 1, membership in NP, will not figure prominently into our arguments here. Requirement 2 is traditionally proven via transitivity: if there exists a polynomial time reduction to B from some language A already known to be in NPC, then because (by definition) all problems in NP reduce in polynomial time to A, all problems in NP reduce in polynomial time to B. In other words, another way of proving requirement 2 is to prove $\exists A \in \text{NPC}, A \leq_{\text{P}} B$.[4] This idea of a *reduction function* (or simply *reduction*) is formalized as

$$\exists f : \Sigma^* \to \Sigma^*, w \in A \iff f(w) \in B \tag{3}$$

Because NPC is concerned with time-complexity, there is an additional requirement that the function f is computable in polynomial time. If this reduction exists, then, because NPC is an equivalence class (with respect to reciprocal polynomial reducibility), there will also exist a second (distinct) polynomial-time reduction g:

$$\exists g : \Sigma^* \to \Sigma^*, w \in B \iff g(w) \in A \tag{4}$$

Both reductions play important roles for different reasons. Given a language B suspected to be NP-complete, a reduction f from a known NP-complete language A is important in proving B is NP-complete. But it is the reduction g that allows existing approximation and solution algorithms for deciding A to be used to decide B.

For the purposes of illustration, let us imagine that we have not yet determined whether or not the CLIQUE problem is NP-complete and that we want to prove that it is. Let us assume we have shown CLIQUE∈NP (satisfying requirement 1). All that remains is to show a valid reduction from an existing NP-complete problem, e.g., 3SAT. As any computational theorist will attest, this is a scenario in which a fair amount of creativity (in the CC sense of the word) must be employed: finding a valid reduction from one NP-complete problem to another. Algorithm 1 from (Sipser 2013) gives pseudocode for a reduction 3SAT $\leq_{\text{P}}$ CLIQUE, and Figure 2 shows the output (2b) of Algorithm 1 for the input (2a). In this example, the string $w = \langle \phi \rangle$ for ϕ shown in Figure 2a, and, because there is a satisfying truth assignment for ϕ (i.e., $x =$ FALSE, $y =$ TRUE), $\langle \phi \rangle \in$ 3SAT. $f(w) = \langle G, k \rangle$ for (G, k) shown in Figure 2b, and because there is a k-clique in G [i.e., $(y_3, \overline{x}_4, \overline{x}_7)$], $\langle G, k \rangle \in$ CLIQUE, as required.

It is worth pausing to note some details about this reduction. Both the 3SAT instance (2a) and the equivalent CLIQUE instance (2b) have modular elements that are parallel between them. For each clause in the 3SAT instance there is a corresponding subgrouping of 3 nodes in the CLIQUE instance (as reflected by the colored highlights). For each Boolean literal in the 3SAT instance there is a corresponding node in the CLIQUE instance. These modular elements and groupings are found in every NP-complete problem and are commonly referred to as *gadgets*. Identifying both the quantity and nature of gadgets in an NP-complete problem is an important first step towards finding a valid reduction because ultimately a reduction is simply a matter of mapping the right gadgets (or some creative combination of gadgets) to one another. In this sense, one can think of NP-complete reductions as a form of analogical reasoning. Here then is a second scenario in which creativity must be employed: creating gadgets for NP-complete problems for use in reductions.

In addition to proving CLIQUE NP-complete, a reduction from 3SAT to CLIQUE also has a practical usage: it allows instances of 3SAT to be solved by existing solution algorithms for CLIQUE.[5] This is a remarkable and useful property of NP-complete problems that is surprisingly underappreciated. In short, rather than having to design, implement, and compare new solutions every time a new NP-complete problem B is discovered, one need simply reduce B to an existing NP-complete problem A and then apply and compare any number of existing solutions to A (or via transitivity to any other NP-complete problem for which reductions from A are available). This application of NP-complete reductions for leveraging existing solutions to NP-complete problems is of significant interest and is a topic we return to below.

Note finally that the reduction shown in Algorithm 1 is only one of an infinite number of valid reduction functions from 3SAT to CLIQUE. In addition, the reduction function itself is incomplete without an accompanying proof that the reduction is in fact a valid mapping reduction and that it is a polynomial-time function.

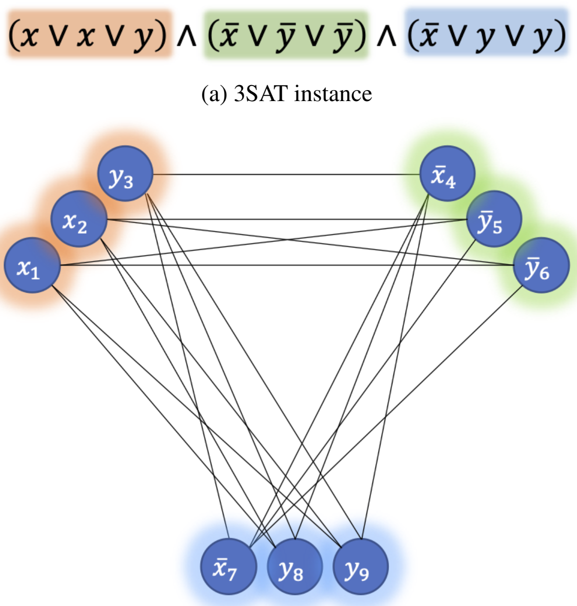
[3] In fact, it has been suggested that the problem of (computational) creativity itself is at least NP-hard, and may very likely be undecidable (Ventura 2014).

[4] This formulation presents a chicken-and-egg conundrum—from where do we get the first NPC problem? The conundrum is resolved with the the Cook-Levin Theorem (Cook 1971), which established Boolean satisfiability (SAT) as that problem by giving an elegant proof that $\forall A \in \text{NP}, A \leq_{\text{P}}$ SAT.

[5] Technically *solving* an NP-complete problem implies finding an optimal solution, but where such is impractical for NP-complete problems, the term *solve* usually refers to the use of heuristic or approximation algorithms to find good, but nonetheless suboptimal, solutions in a more tractable time frame.

Algorithm 1 Reduction from 3SAT to CLIQUE

```
Require: A 3CNF Boolean expression $\phi$
procedure REDUCE($\phi$)
    $N \leftarrow \{\lambda_i | \lambda_i$ is the $i$th instance of literal $\lambda$ in $\phi\}$
    $V \leftarrow \{(\lambda_i, \nu_j) | \lambda_i, \nu_j \in N$
                and $\lambda_i, \nu_i$ are not in the same clause in $\phi$
                and $\lambda \neq \overline{\nu}\}$
    $G \leftarrow (N, V)$
    $k \leftarrow$ the number of clauses in $\phi$
    return $G, k$
```


$(x \vee x \vee y) \wedge (\bar{x} \vee \bar{y} \vee \bar{y}) \wedge (\bar{x} \vee y \vee y)$
y_3
x_2
x_1
$\bar{x}_4$
$\bar{y}_5$
$\bar{y}_6$
$\bar{x}_7$
y_8
y_9

(a) 3SAT instance

(b) CLIQUE instance ($k = 3$)

Figure 2: *3SAT to CLIQUE reduction.* (a) an instance of the 3SAT problem and (b) equivalent CLIQUE instance to which the 3SAT instance reduces. Matching clause gadgets are highlighted with colors. Both the function (Algorithm 1) that maps the 3SAT instance to the CLIQUE instance as well as the individual gadgets in the generated CLIQUE instance represent artifacts resulting from creative processes.

Analogical reasoning

The concept of using a reduction f to compare an instance of problem A to an equivalent instance of problem B is, in some sense, a formalization of analogical reasoning: a is to A as b is to B. Finding f is essentially finding the relationship that makes the analogy valid. While this form of analogy has not yet been addressed in the CC literature, there has been work on other forms, including lexical analogy using WordNet (Hayes, Veale, and Seco 2004); bilingual lexical analogy using HowNet, an ontology for English and Chinese (Veale 2006); cross-domain analogical reasoning for improved text generation (Hervás et al. 2006); analogy emerging as a consequence of concept space exploration (Thornton 2008); constructing visual analogies of the kind found on intelligence tests (McGreggor, Kunda, and Goel 2010); analogical reasoning for mathematical creativity (Pease, Guhe, and Smaill 2010); using analogy for story generation (Zhu and Nón 2010); an autonomous system for generating analogical comparisons (O'Donoghue and Keane 2012); analogy to facilitate concept-blending (Besold and Plaza 2015); and transforming song lyrics using vector-based analogy in word embeddings (Oliveira 2020).

Four CC Problems in NP-completeness Theory

Having outlined the basic concepts relevant to NP-completeness, we can now identify four open questions/problems in this area that are ideally suited for being addressed by CC systems:

1. Given NP-complete problems A and B, can we create a valid polynomial-time reduction from A to B?
2. Given an NP-complete problem A, can we define meaningful gadgets for A that would be helpful in creating a valid polynomial-time reduction to/from A?
3. There are many examples of games/puzzles that are NP-complete. Given an NP-complete problem A and an NP-complete game/puzzle G, can we either create a new reduction or modify an existing reduction from A to G such that the reduced game/puzzle instances of G are fun/engaging?
4. Given an NP-complete problem A, can we create an efficient, effective polynomial-time heuristic or approximation algorithm to solve A?

Note that only the last of these proposed artifacts represents an actual (approximate) solution to an NP-complete problem. While creating a system that produces algorithmic (approximate) solutions to arbitrary NP-complete problems has not yet been addressed directly in the CC literature, there has been some work on CC systems/approaches for producing computer programs to solve arbitrary problems (Cook, Colton, and Gow 2013; Charnley et al. 2016; Znidarsic et al. 2016; Colton, Powley, and Cook 2018; Colton et al. 2019), and, to our knowledge, this is the only one of the four questions that has been given previous consideration in the CC literature.[6] So, although we include it as an example of a CC problem from the domain of CCT, we recognize that CC for computer programming is of much broader interest and in some sense its own subdomain of CC. And as well it should be; while the first three problems are well-defined in terms of the typicality constraints they must satisfy, the intention that they must meet, and the value they should provide, the creation of arbitrary computer programs for solving arbitrary problems is much less well-defined. For these reasons, we will focus the following discussion on the first three problems and direct the reader to extant literature that addresses the fourth.

Here we consider each of the first three questions/problems in more detail. We define each as a decision

[6] Recently, some work has also been done on the reverse problem—applying software engineering principles to the problem of designing CC systems (Glines, Griffith, and Bodily 2021).

problem and discuss notions of typicality, novelty, intentionality, and value in each of the three domains.

Artifact 1: NP-Complete Reduction Algorithms

Given NP-complete problems A and B, can we create a valid polynomial-time reduction from A to B? This question represents an open and challenging problem in CCT that essentially presents the set of all valid polynomial-time reductions as (potentially) creative artifacts. Represented as a decision problem, this set could be written as

$$\begin{aligned}\text{REDUCTIONS} = \{\langle A, B, f\rangle \mid\ & A, B \in \text{NPC and} \\ & f \text{ is a polynomial-time} \\ & \text{reduction from } A \text{ to } B\}\end{aligned} \tag{5}$$

In order to meet standards of typicality for artifacts in this domain, a reduction f must meet at least two basic criteria: first, f must be a valid reduction from A to B—that is, it must be proven that $w \in A \iff f(w) \in B$; second, a reduction must operate in polynomial-time. Besides being necessary for typicality, a well-formed proof demonstrates intentionality in the reduction artifact.

In the authors' personal experience, the creation of a reduction function is an iterative process that starts simply with finding functions that translate a well-formed instance of problem A into a well-formed instance of problem B followed by experimentation and (if experimentation is successful) a formal proof. If experimentation is unsuccessful, the function is revamped. In light of this, even a system that is capable of translating a well-formed instance of problem A into a well-formed instance of problem B would possess significant value as a co-creative agent.

Novelty in this domain is primarily a function of whether *any* reduction from A to B has been previously documented. CCT guarantees that there exists a valid polynomial-time reduction between every pair of NP-complete problems. However, compared to the vast number of reductions that we know exist, relatively few reductions have been published. Several efforts have been undertaken to catalog what reductions have been published. The Redux platform discussed below represents an effort currently underway to make such reductions more accessible via Web APIs and a pedagogical visualization tool. Where a reduction has been published for a particular A, B pair, novelty can still be measured by comparing the similarities/differences between the several reductions that have been presented.

In assessing the value of a particular reduction, there are a few characteristics worth considering. First, valued reductions tend to be those which reduce instances of A to simpler (i.e., smaller) instances of B. For example, if one 3SAT-CLIQUE reduction algorithm reduces a 3CNF Boolean formula ϕ with k clauses to a graph with $3k$ nodes and a second reduction reduces ϕ to a graph with $4k$ nodes, we would value the simpler graph (all else held equal). Second, valued reductions are explainable reductions. Explainability is a metric that has previously been suggested for assessing value (Bodily and Ventura 2018).

Artifact 2: NP-Complete Reduction Gadgets

Given an NP-complete problem A, can we define meaningful gadgets for A that would be helpful in creating a valid polynomial-time reduction to/from A? This question essentially presents the set of all possible gadgets for a problem as (potentially) creative artifacts. The notion of what defines a gadget is inherently ambiguous as it varies from one problem to another and depends on whether the problem is on the input or output end of the reduction. This is a domain that will be easier to define as more and more examples of gadgets are cataloged. In general, we can think of a gadget $t(w)$ as a function that, given an instance w of an NP-complete problem, returns some collection of subunits of w. Represented as a decision problem, we could write this set as

$$\begin{aligned}\text{GADGETS} = \{\langle A, t\rangle \mid A \in \text{NPC and } \forall w \in A, t(w) \\ \text{generates a collection of subunits of } w\}\end{aligned} \tag{6}$$

We have seen how gadgets are valuable for the role they play in reductions. However, it is likely that gadgets could have value in other contexts, as well. For example, consider the goal of designing an algorithm that, given an NP-complete problem A, generates a greedy heuristic algorithm to solve A. Many such algorithms consist of little more than a few nested while loops iterating over what essentially amount to gadgets (e.g., a greedy heuristic algorithm for 3SAT would likely involve some sort of loop over clauses with an inner loop over variables within the clause). In general, we consider that defining the concept of a gadget for a particular problem has the potential of being a valuable creative artifact independent from whatever context in which it might be used.

With this in mind, intentionality in the definition of gadgets could be fixed on their intended use. When gadgets are intended for use in designing reduction functions, their value would depend on whether or not they contribute to a valid reduction. Again, simple gadgets are (all else held equal) valued over more complex gadgets. Whereas explainability serves as a meaningful value metric for reductions, it is sometimes more difficult to make an argument for this metric with respect to gadget artifacts, though certainly, gadgets that are intuitive or that do elucidate the construction or interpretation of a reduction will be of high value.

The novelty of a gadget not only depends on the definition of the gadget itself but also on the context in which it is used. For a given problem A, different gadgets for A become useful in reductions to/from different problems so that the presentation of a particular gadget when constructing a new reduction to/from a problem B could be considered a form of novelty.

In general, a typical gadget is some atomic unit of a problem instance and typically gadgets are exact subsequence/subset/subgraph units of an instance.

Artifact 3: NP-Complete Game Instances

Given an NP-complete problem A and an NP-complete game/puzzle G, can we either create a new reduction or modify an existing reduction from A to G such that the reduced game/puzzle instances of G are fun/engaging? We could, of course, consider instead the problem of simply trying to make game/puzzle instances of an NP-complete problem A more creative. But again, this is not particularly unique to CCT (many researchers have considered the challenge of making games more creative). Far more interesting and specific to CCT is consideration of how to amplify creativity in games/puzzles *that are formed as reductions from other NP-complete problems*. Represented as a decision problem, we could write this set as

$$\begin{aligned}\text{GAME} = \{\langle A, G, f\rangle \mid\ & A \in \text{NPC and} \\ & G \in \text{NPC is a game or puzzle} \\ & \text{and } f \text{ is a polynomial-time} \\ & \text{reduction from } A \text{ to } G\}\end{aligned} \quad (7)$$

This problem is what initially piqued our interest in applying CC to CCT: could we take an arbitrary NP-complete problem from the real world and turn it into a game or puzzle that people would find engaging? Human intuition is remarkably adept at finding good solutions to NP-complete problems. Unfortunately, people do not typically enjoy solving Boolean satisfiability or graph theory problems. But they do like games. If we can render arbitrary NP-complete problems as fun and engaging games or puzzles then we can leverage the power of crowd-sourcing to find solutions that may be better than any computer could come up with (Cusack et al. 2010).

As an example, consider the protein folding game FoldIt (Cooper et al. 2010). According to its website,

> *Knowing the structure of a protein is key to understanding how it works and to targeting it with drugs. The number of different ways even a small protein can fold is astronomical because there are so many degrees of freedom. Figuring out which of the many, many possible structures is the best one is NP-complete and is regarded as one of the hardest problems in biology today. Current methods take a lot of money and time, even for computers. Foldit attempts to predict the structure of a protein by taking advantage of humans' puzzle-solving intuitions and having people play competitively to fold the best proteins*

See Figure 3 for an example screenshot of the game.

Our initial foray into this problem was an attempt to reduce the well-known NP-complete travelling salesperson problem to a popular NP-complete flood fill game called KAMI. We know from CCT that a reduction exists. However, despite months of trying, we have yet to devise a valid solution (much of this time was spent creating and combining different gadget artifacts from these two problems). We are aware of a reduction from the shortest common supersequence problem to flood fill puzzles which provided some ideas on how gadgets could be created for the KAMI problem (see Figure 4) (Marchetti and Bodily 2022; Clifford et al. 2012).

Many games and puzzles have been shown to be NP-complete including well-known examples such as *Battleship* (Sevenster 2004), *FreeCell* (Helmert 2003), *Instant Insanity* (Garey and Johnson 1979), *LaserTank* (Alexandersson and Restadh 2020), *Pandemic* (Nakai and Takenaga 2012), *Rubik's Cube* (Demaine, Eisenstat, and Rudoy 2018) and *Sudoku* (Yato and Seta 2003). Because these games are well-known and fun, they are excellent candidates for reduction targets from other NP-complete problems.

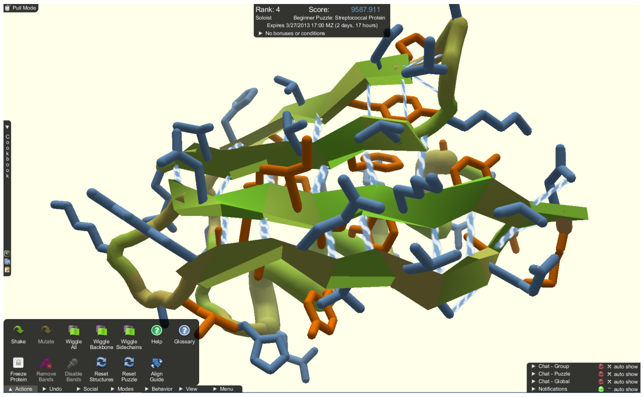


Figure 3: *FoldIt*. FoldIt is a crowd-sourced game for solving difficult instances of the protein folding problem, an NP-complete problem from biology, the solutions to which have implications in understanding protein function (and thus also for designing medical or other interventions). Screen shot from the game taken from `https://fold.it`.

As far as what defines creativity in this domain, since art and gaming already enjoy broad treatment in the CC field, more traditional definitions of novelty, value, typicality and intentionality can be invoked directly to assess the quality of this artifact type. Although the definitions of GAME and REDUCTION are very similars, in the case of REDUCTION the focus is on the behavior of creating novel, valid reductions *ab initio*. In the case of GAME, we take a valid reduction for granted and focus on the creativity of the artifacts generated *from* the reduction. One possible approach to attacking this problem may be related to procedural puzzle generation (De Kegel and Haahr 2020).

There are two concerns that should be mentioned with regard to reducing NP-complete problems to CC-enhanced games. First, most NP-complete games are played with instances of very limited input sizes. An instance of the game Battleship, for example, is defined in terms of the size of the board (typically 10×10) and the number of ships (typically 5). One can easily imagine reductions from arbitrary NP-complete problem instances that could result in very large game instances (imagine playing Battleship on a $10,000 \times 10,000$ board with 5,000 ships), much larger than human players are used to playing and larger perhaps than would appeal to many players. This diminishing value with increasing input size is certainly relevant to considerations on how CC might be used to attempt to create valuable artifacts in this space. It is worth noting that the FoldIt game (Figure 3) is at least one example of an NP-complete game with non-trivially-sized instances that has seen success.

Second, reduction algorithms tend to be highly prescriptive which could severely limit the variability with which game instances could be rendered. For example, KAMI has

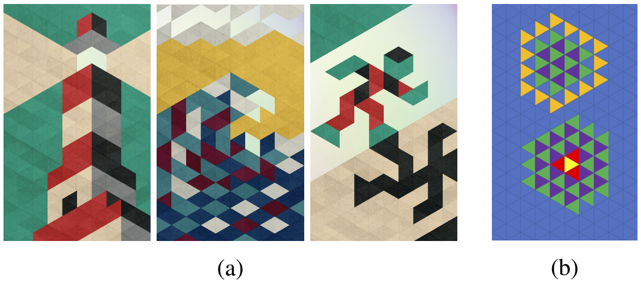


Figure 4: *Limitations on creativity in derived KAMI puzzles.* (a) KAMI is an NP-complete flood fill game whose puzzles typically allow a wide range of aesthetic expression. (b) Due to the highly prescriptive nature of reduction algorithms, a KAMI puzzle that is derived via reduction from an instance of the NP-complete shortest common supersequence problem will always necessarily be composed of diamond gadgets, like those shown, which significantly constrains the ways in which CC could be applied to enhance the creativity of the puzzle without invalidating the reduction.

come to be known for its highly aesthetic and creative puzzles (e.g., see Figure 4a). However, when we take a different NP-complete problem, e.g., the shortest common supersequence (SCS) problem, and reduce it to KAMI, the result is always a puzzle consisting of several diamonds (the diamond is a gadget), each consisting of a pattern of concentric multicolored rings (color is another gadget), all on a common background canvas color (see Figure 4b). The number, size, color scheme, and even shape (to some extent) of the diamonds could change without invalidating the reduction, but otherwise, all KAMI puzzles generated from a particular SCS reduction will follow a similar pattern (Clifford et al. 2012). It is possible that other reductions could potentially produce other patterns. The point being made here is that the nature of reductions is highly prescriptive and consequently places some limits on *how* CC would need to be applied to enhance the creativity of puzzles derived from NP-complete reductions in order not to invalidate the reductions.

An Ontology of NP-completeness

CC systems across a spectrum of application domains rely on knowledge bases of existing artifacts from which they extract patterns for the creation of new artifacts (Ventura 2017). Though some efforts have been made to create a knowledge base of NP-complete problems, there does not exist a well-established resource cataloging NP-complete problems, reductions, and/or solutions. To this end we have undertaken to create Redux, an ontological knowledge base of NP-complete problems, reductions, solutions, and verifiers accessible via a web API[7]. In addition to the knowledge base, we also aim to build a pedagogical visualization front end to the knowledge base. A mockup of the system can be seen in Figure 5.

We envision the Redux knowledge base allowing researchers to perform meta-analyses over various aspects of NP-complete problems in order to gain insights on such questions as:

- What gadgets have been identified/created before when reducing to/from a particular NP-complete problem?
- What patterns exist in how gadgets for one problem map to gadgets for another problem?
- What in general is the relationship between previously identified gadgets for a particular NP-complete problem and the formulation of, say, a greedy heuristic solution algorithm for the problem?
- What additional power or knowledge can be leveraged via transitivity of the reductions in the knowledge base?

In addition, researchers will be able to directly access NP-complete problem definitions, example instances, and formatting; call reduction algorithms between particular pairs of NP-complete problems; run solution algorithms for particular NP-complete problems; and verify proposed solutions for particular NP-complete problem instances. Our hope is that this knowledge base will spur innovative ideas and solutions in and around the domain of NP-completeness.

Conclusion

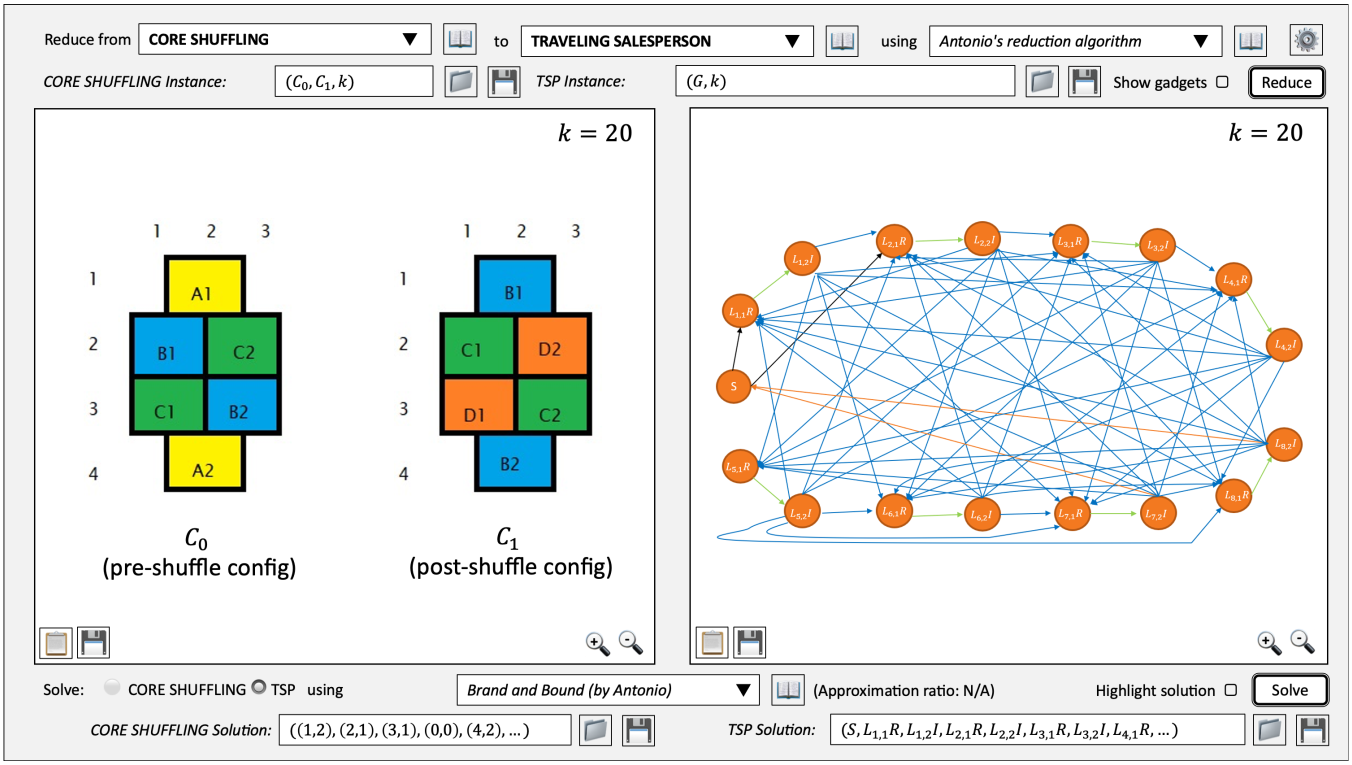
The CC research community has long been interested in balancing its focus on applications in artistic domains with more CC applications in the fields of science, mathematics, and logic. Our purpose in this paper has been to suggest that computational complexity theory, and NP-completeness in particular, is a ripe candidate for contributing to this balance. We have attempted to highlight and provide some definition to four open CC problems in this domain. We have argued that progress towards addressing these problems promises to make significant impacts in the field of CCT, which in turn promises to make significant impacts in many real-world domains.

In addition we have presented Redux, a nascent ontological knowledge base of NP-complete problem definitions, reductions, and solution algorithms. Our intent is to augment the knowledge base through crowd-sourcing with the ultimate goal of providing a comprehensive and accessible resource on NP-completeness by which CC and other researchers can push forward the boundaries of applied computational complexity research. As a final note, it is worth mentioning that many of the significant open problems in applying CC are themselves likely NP-complete (or harder) problems. And though it is also likely that creativity itself lies beyond the realm of NP-completeness, advances in CCT are likely to translate directly into advances in the field of computational creativity.

Author Contributions

Both authors contributed to all aspects of the work, including ideation, narrative/position development and writing.

[7] http://redux.aws.cose.isu.edu/



Reduce from CORE SHUFFLING to TRAVELING SALESPERSON using *Antonio's reduction algorithm*

CORE SHUFFLING Instance: (C_0, C_1, k)

TSP Instance: (G, k)

Show gadgets □

Solve: ○ CORE SHUFFLING ● TSP using *Brand and Bound (by Antonio)* (Approximation ratio: N/A)

Highlight solution □

CORE SHUFFLING Solution: ((1,2), (2,1), (3,1), (0,0), (4,2), ...)

TSP Solution: $(S, L_{1,1}R, L_{1,2}I, L_{2,1}R, L_{2,2}I, L_{3,1}R, L_{3,2}I, L_{4,1}R, \ldots)$

Figure 5: *The Redux application*. The tool shown serves as a graphical user interface providing access to a crowd-sourced knowledge base of NP-complete problems, reduction algorithms, and solution algorithms. Users can select (or add new) problems. A unique reduction algorithm is required for each unique (ordered) pair of selected problems. Users can contribute new reduction algorithms. Then for a given instance of the problem on the left [e.g., (nuclear) CORE SHUFFLING], the reduction algorithm is applied to generate an equivalent instance of the problem on the right (e.g., TRAVELING SALESPERSON). A solution to one problem instance can also be mapped to the equivalent solution for the equivalent problem instance. Visualization of instances with gadgets and/or solutions highlighted is included for pedagogical purposes. The tool highlights the power of reduction, allowing existing solutions to one problem to be reused to solve new problems. Possible applications of CC in this context include using CC to create novel reduction algorithms; using CC to propose gadgets for co-creative development of reduction algorithms; application of CC to aesthetically present reduced problem instances as engaging puzzles pursuant to crowd-sourcing solutions to NP-complete problems; and using CC to create novel solution algorithms.

Acknowledgments

The authors gratefully acknowledge Antonio Tahhan for contribution of the core shuffling diagrams and Mark Bodily for the contribution of the KAMI puzzle examples.

References

Alexandersson, P., and Restadh, P. 2020. Lasertank is NP-complete. In *Mathematical Aspects of Computer and Information Sciences*, LNCS 11989, 333–338.

Besold, T. R., and Plaza, E. 2015. Generalize and blend: Concept blending based on generalization, analogy, and amalgams. In *Proceedings of the Sixth International Conference on Computational Creativity*, 150–157.

Bodily, P., and Ventura, D. 2018. Explainability: An aesthetic for aesthetics in computational creative systems. In *Proceedings of the 9th International Conference on Computational Creativity*, 153–160.

Charnley, J.; Colton, S.; Llano, M. T.; and Corneli, J. 2016. The FloWr online platform: Automated programming and computational creativity as a service. In *Proceedings of the 7th International Conference on Computational Creativity*, 363–370.

Clifford, R.; Jalsenius, M.; Montanaro, A.; and Sach, B. 2012. The complexity of flood filling games. *Theory of Computing Systems* 50(1):72–92.

Colton, S.; Pease, A.; Cook, M.; and Chen, C. 2019. The HR3 system for automatic code generation in creative settings. In *Proceedings of the 10th International Conference on Computational Creativity*, 108–115.

Colton, S.; Powley, E. J.; and Cook, M. 2018. Investigating and automating the creative act of software engineering: A position paper. In *Proceedings of the 9th International Conference on Computational Creativity*, 224–231.

Cook, M.; Colton, S.; and Gow, J. 2013. Nobody's a critic: On the evaluation of creative code generators—a case study in video game design. In *Proceedings of the 4th Interna-*

tional Conference on Computational Creativity, 123–130.

Cook, S. 1971. The complexity of theorem proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, 151–158.

Cooper, S.; Khatib, F.; Treuille, A.; Barbero, J.; Lee, J.; Beenen, M.; Leaver-Fay, A.; Baker, D.; Popović, Z.; et al. 2010. Predicting protein structures with a multiplayer online game. *Nature* 466(7307):756–760.

Cusack, C.; Largent, J.; Alfuth, R.; and Klask, K. 2010. Online games as social-computational systems for solving NP-complete problems. In *Meaningful Play*. Michigan State University East Lansing.

De Kegel, B., and Haahr, M. 2020. Procedural puzzle generation: A survey. *IEEE Transactions on Games* 12(1):21–40.

Demaine, E.; Eisenstat, S.; and Rudoy, M. 2018. Solving the Rubik's cube optimally is NP-complete. In *Proceedings of the 35th Symposium on Theoretical Aspects of Computer Science*, article 24.

Garey, M. R., and Johnson, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: W. H. Freeman.

Glines, P.; Griffith, I.; and Bodily, P. 2021. Software design patterns of computational creativity: A systematic mapping study. In *Proceedings of the 12th International Conference on Computational Creativity*, 218–221.

Hayes, J.; Veale, T.; and Seco, N. 2004. The Bible is the Christian-Koran: Exploring lexical analogy via WordNet. In *Proceedings of the First Joint Workshop on Computational Creativity*, 59–64.

Helmert, M. 2003. Complexity results for standard benchmark domains in planning. *Artificial Intelligence* 143(2):219–262.

Hervás, R.; Pereira, F. C.; Gervás, P.; and Cardoso, A. 2006. Cross-domain analogy in automated text generation. In *Proceedings of the Third Joint Workshop on Computational Creativity*.

Loughran, R., and O'Neill, M. 2017. Application domains considered in computational creativity. In *Proceedings of the 8th International Conference on Computational Creativity*, 197–204.

Marchetti, K., and Bodily, P. M. 2022. KAMI: Leveraging the power of crowd-sourcing to solve complex, real-world problems. In *Proceedings of the 2nd Intermountain Engineering, Technology, and Computing Conference*.

McGreggor, K.; Kunda, M.; and Goel, A. 2010. A fractal approach towards visual analogy. In *Proceedings of the First International Conference on Computational Creativity*, 65–74.

Nakai, K., and Takenaga, Y. 2012. NP-completeness of Pandemic. *Journal of Information Processing* 20(3):723–726.

Oliveira, H. G. 2020. Weirdanalogymatic: Experimenting with analogy for lyrics transformation. In *Proceedings of the Eleventh International Conference on Computational Creativity*, 228–235.

O'Donoghue, D., and Keane, M. T. 2012. A creative analogy machine: Results and challenges. In *Proceedings of the Third International Conference on Computational Creativity*, 17–24.

Pease, A.; Colton, S.; Warburton, C.; Nathanail, A.; Preda, I.; Arnold, D.; Winterstein, D.; and Cook, M. 2019. The importance of applying computational creativity to scientific and mathematical domains. In *Proceedings of the 10th International Conference on Computational Creativity*, 250–257. Association for Computational Creativity.

Pease, A.; Guhe, M.; and Smaill, A. 2010. Some aspects of analogical reasoning in mathematical creativity. In *Proceedings of the First International Conference on Computational Creativity*, 60–64.

Sevenster, M. 2004. Battleships as a decision problem. *ICGA Journal* 27(3):142–149.

Sipser, M. 2013. *Introduction to the Theory of Computation*. Boston: Cengage Learning.

Thornton, C. 2008. Analogy as exploration. In *Proceedings of the Fifth Joint Workshop on Computational Creativity*, 81–90.

Veale, T. 2006. Re-representation and creative analogy: A lexico-semantic perspective. *New Generation Computing* 24(3):223–240.

Ventura, D. 2014. Can a computer be lucky? and other ridiculous questions posed by computational creativity. In *Proceedings of the Seventh Conference on Artificial General Intelligence*, LNAI 8598, 208–217.

Ventura, D. 2017. How to build a CC system. In *Proceedings of the 8th International Conference on Computational Creativity*, 253–260.

Yato, T., and Seta, T. 2003. Complexity and completeness of finding another solution and its application to puzzles. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* E86-A(5):1052–1060.

Zhu, J., and Nón, S. O. 2010. Towards analogy-based story generation. In *Proceedings of the First International Conference on Computational Creativity*, 75–84.

Znidarsic, M.; Cardoso, A.; Gervás, P.; Martins, P.; Hervas, R.; Alves, A. O.; Oliveira, H.; Xiao, P.; Linkola, S.; Toivonen, H.; Kranjc, J.; and Lavrac, N. 2016. Computational creativity infrastructure for online software composition: A conceptual blending use case. In *Proceedings of the 7th International Conference on Computational Creativity*, 371–379.

Toward Life-Long Creative Problem Solving: Using World Models for Increased Performance in Novelty Resolution

Evana Gizzi[1], Wo Wei Lin[2], Mateo Guaman Castro[3], Ethan Harvey[1], Jivko Sinapov[1]
[1] Department of Computer Science, Tufts University, MA, USA
[2] Department of Electrical and Computer Engineering, Tufts University, MA, USA
[2] The Robotics Institute, Carnegie Mellon University, PA, USA
{Evana.Gizzi,Wo_Wei.Lin,Ethan.Harvey,Jivko.Sinapov}@tufts.edu, mguamanc@andrew.cmu.edu

Abstract

Creative problem solving (CPS) is a skill which enables innovation, often times through repeated exploration of an agent's world. In this work, we investigate methods for life-long creative problem solving (LLCPS), with the goal of increasing CPS capability over time. We develop two world models to facilitate LLCPS which use sub-symbolic action and object information to predict symbolic meta-outcomes of actions. We experiment with three CPS scenarios run sequentially and in simulation. Results suggest that LLCPS is possible through the use of a world model, which can be trained on CPS exploration trials, and used to guide future CPS exploration.

Introduction

Creative problem solving (CPS) is a skill which enables adaptation to novel situations, often through innovating on-the-fly (Gizzi et al., 2020, 2022). A key process in creative problem solving work is an agent's exploration with its environment, which typically requires many interactions in order to find a solution to encountered novelty. To date, research in CPS within the field of artificial intelligence has focused predominantly on resolving a singular novel task-at-hand. For example, when a robot needs to figure out how to push a heavy object, it may explore alternative parameterizations of a `push` actions to discovery a `strike` action. In this circumstance, the agent will explore until a solution is found. In doing so, these exploration trials are often "disposed of" in CPS resolution. These interim exploration episodes typically contain a large number of agent-environment interactions, which provide a large and fruitful data sample of experience for the agent to otherwise learn from.

In this paper, we develop a method for enabling *life-long creative problem solving (LLCPS)*, which uses CPS exploration data to train a world model to increase CPS performance over time. The world model is continuously trained on both a) past CPS exploration trials and b) any past world interactions. We train two world models (a neural network and a naive Bayes model) with a combination of sub-symbolic and symbolic data as input, and symbolic data as the output. In doing so, we are able to direct the agent in its CPS exploration to avoid those trials which are not likely to resolve the CPS task, which decreases the total amount of exploration in CPS over time. We evaluated our approach in a 3D physics-based simulation environment, over three consecutive experimental scenarios to observe how CPS performance changes over time, and compared our approach to three alternative baseline world model choices.

Related Work

Although life-long creative problem solving has not been directly explored in research, similar lines of work investigate *life-long learning*, which develops methods to enable continual learning over time such that an agent is able to utilize *both* its past experiences and new knowledge (see Parisi et al. (2019) for a review). For example, Rao et al. (2019) develop a custom model for continual life-long learning which leverages a suit of artificial intelligence methods to learn representations of tasks on the fly without labels or human intervention. Within the mobile robotics navigation domain, Kahn et al. (2021) develop a method which gathers data for off-policy training of a retroactive self-supervised predictive model, centered around environment affordance learning. *Multi-task learning (MTL)* is an area within machine learning that aims to learn a general task model using samples from multiple tasks as a way to derive shared representation (Crawshaw, 2020). In doing so, MTL aims to address the data efficiency issues that are typical in machine learning for single task learning (STL), to increase performance in learning – but not necessarily to specifically be used for novel problem solving. For example, in Kalashnikov et al. (2021), a generalized multi-task deep reinforcement learning method called "MT-Opt" is trained off-line, simultaneously across multiple robot manipulation tasks. Similarly, *meta-reinforcement learning (MRL)* aims to increase performance in general task handling by optimizing adaptation to new tasks (Yu et al., 2020). For example, in Javed and White (2019), a meta-level objective is used in MRL to minimize catastrophic interference and promote future learning via naturally sparse representation learning. Unlike MTL, MRL assumes that all training and testing (novel task) data is drawn from the same task distribution.

Theoretical Framework

Consider an agent which is able to act in its world through *symbolic* planning to as a method for accomplishing tasks.

Additionally, the agent is able to use perceived *sub-symbolic* information about its world in order to learn a *world model* to resolve novelty in task failure.

Symbolic Planning

We assume that the robot has a symbolic knowledge base $\mathcal{K}$, defined as a classical planning problem, where $\mathcal{K} = \langle \mathcal{S}_{\mathcal{K}}, \mathcal{A}_{\mathcal{K}}, \mathcal{E}_{\mathcal{K}}, \mathcal{P}_{\mathcal{K}} \rangle$, with respective components denoted $\mathcal{S}, \mathcal{A}, \mathcal{E}, \mathcal{P}$ for brevity. The set $\mathcal{S}$ indicates possible world states, reachable by taking *actions* on *entities* (either manipulable, static, or parts of the agent) in the sets $\mathcal{A}$ and $\mathcal{E}$ respectively. Specifically, $\mathcal{S} = \{s_1 \dots s_n\}$, $\mathcal{E} = \{e_1 \dots e_p\}$, and $\mathcal{A} = \{a_1(\triangledown_1) \dots a_m(\triangledown_m)\}, \triangledown_i \subseteq \mathcal{E}$, where the elements in a ordered list $\triangledown_i$ are considered to be the *arguments* of its corresponding action a_i. Note that in general, entities can be physical objects in the environment or the end effectors of the robot, but in this work we only consider physical manipulable objects in the environment. We define a set of known predicate descriptors, or ***fluents***, which can be used to describe entities in the world as $\mathcal{F} = \{f_1(\triangledown) \dots f_q(\triangledown)\}$ along with their negations $\hat{\mathcal{F}} = \{\hat{f_1(\triangledown)} \dots \hat{f_q(\triangledown)}\}$, where $\triangledown \subset \mathcal{E}$. Together, the predicate descriptors and their negations comprise an encompassing set of predicates $\mathcal{P} = \mathcal{F} \bigcup \hat{\mathcal{F}}$ which is used by the agent to describe states, entities, and information about the execution of actions, as is typical in planning domains. Thus, a given state $s_i \in \mathcal{S}$ is composed of a finite set of predicates $\mathcal{F}_i \subset \mathcal{F}$ which hold true in world state s_i. Note, this does not include negation predicates in $\mathcal{F}$, although these may be deduced by the planning agent. Moreover, we assume a *planning domain definition language (PDDL)* representation of actions, where each action has a set of *preconditions* and *effects*, denoted ρ_i, $p_i \in \mathcal{P}$, indicating the predicates which must hold true *before* executing an action (preconditions), and those which are assumed to hold true *after* executing an action (effects). Note that the preconditions and effects can include those negation predicates in $\hat{\mathcal{F}}$, described earlier.

The agent is able to use the aforementioned information to act in its world, through planning, to accomplish tasks. We define a task $\mathcal{T}$ in $\mathcal{K}$ as $\mathcal{T} = (\mathcal{K}, s_0, s_g)$, where s_0 is an initial state, s_g is a goal state, and $s_0, s_g \in \mathcal{S}$ (recall a state is composed of a set of fluents which hold true). A plan $\pi = [a_1, \dots a_{|\pi|}]$ is a solution to accomplishing task $\mathcal{T}$.

Sub-symbolic-based Learning

Next, we describe the sub-symbolic information known and perceivable to the agent. For a given symbolic knowledge base $\mathcal{K}$, we assume that the robot has a corresponding sub-symbolic knowledge base Ψ, containing low-level action executors and object feature information (collectively described as the tuple ($\mathcal{K}$, Ψ)). Specifically, $\Psi = \langle \mathcal{R}, X \rangle$, where $\mathcal{R} = \{r_1 \dots r_{|\mathcal{A}_{\mathcal{K}}|}\}$ denotes a set of action controllers for the actions in $\mathcal{A}_{\mathcal{K}}$, and $X = \{x_1 \dots x_{|\mathcal{E}_{\mathcal{K}}|}\}$ denotes a set of feature mappings $x_i : e_i \mapsto \mathbb{R}^n$ for the objects in $\mathcal{E}_{\mathcal{K}}$, where n is the size of the input vector (experimentally chosen), discussed in the next paragraph. For every action in $a_i \in A_{\mathcal{K}}$, there exists a corresponding action controller $r_i \in \mathcal{R}$ which is able to execute a_i with various sub-symbolic parameterizations. Thus $|A_{\mathcal{K}}| = |R|$. Additionally, for every entity $e_i \in \mathcal{E}_{\mathcal{K}}$, there exists a feature mapping $x_i \in X$ which contains sub-symbolic information about entity properties. For every entity list $\mathcal{E}_j$, there exists a list of feature mappings $\hat{X}_j$ which contains the mappings x_i of individual entities in $e_i \in \mathcal{E}_j$.

Value Description (type)	Value Possibilities
encoded action (int)	{1,2,3,4}
rate (float)	action specific
movementMagnitude (float)	action specific
encoded orientation (int)	{1,2}
encoded shape (int)	{1,2,3}
volume (float)	[0.0,∞)
encoded color (float)	{1,2,3}
entity vector magnitude (float)	[0.0,∞)
unit vector x (float)	[0.0,1.0]
unit vector y (float)	[0.0,1.0]
unit vector z (float)	[0.0,1.0]

Table 1: World model $\mathcal{W}$ input values. Data types and value possibilities of each feature in our proof-of-concept is shown. Values which are encoded into numeric values are as follows: action (1 = `push_together`, 2 = `remove_from_container`, 3 = `place_in_container`), orientation (1 = left, 2 = top), shape (1 = sphere, 2 = box, 3 = cylinder), color (1 = red, 2 = blue, 3 = green).

A given feature space X has a cardinality n (denoted $|X|_n$) such that every feature vector mapping $x_i \in X$ is represented as a feature vector containing n distinct object features (thus, $|x_i| = n$). Therefore, for a given knowledge base Ψ, entities can be described using exactly n feature values. Furthermore, we assume that the agent is able to perceive the values of a given feature space through visual or haptic feedback. We assume that the agent starts with all features abstracted already, and thus, in our proof of concept, we do not require the agent to discover these features.

Forward Model We define a world model for our hybrid tuple ($\mathcal{K}$, Ψ) as $\mathcal{W} : (a_i, r_i, \triangledown_i, X_i) \mapsto \Omega$ where Ω defines the static output vector of the world model, which numerically encodes fluent changes which incur after the mapping (See Table 2 for our proof-of-concept world model output choices. Note that the output can be changed to suit the domain). The input to the mapping is a given action a_i with parameter settings r_i, executed over arguments $\triangledown_i$ with corresponding feature vectors X_i (See Table 1 for our proof-of-concept world model input choices. Note that the input can be changed to suit the domain). Thus, for any action, parameter settings to that action, entity arguments to that action, and corresponding feature mappings or the entity arguments, $\mathcal{W}$ is able to predict what fluent states in the world may change as a result of executing a_i on e_i with low-level settings r_i and X_i.

Problem Formulation

Given a task $\mathcal{T}$, a planner generates a plan π to accomplish a goal state s_g. The planning agent, containing an accu-

Value Description	Value Possibilities
positive visibility change	{0,1}
negative visibility change	{0,1}
positive reachability change	{0,1}
negative reachability change	{0,1}
positive touching change	{0,1}
negative touching change	{0,1}

Table 2: World model $\mathcal{W}$ output values. Our proof-of-concept output vector Ω is defined by 6 output values, each characterizing meta-level symbolic changes in the world. A 0 value indicates none of the meta-level changes (in value description) occurred, whereas a 1 indicates 1 or more instances of the meta-level change occurred.

rate representation of the world in its symbolic knowledge base $\mathcal{K}$, is able to successfully execute π, thereby achieving its goal state s_g. We refer to this case as the *original scenario*. Now, suppose that in the case of novelty, something about the world changes such that $\mathcal{K}$ is no longer sufficient, but needs to be updated with new information such that $\mathcal{K}$ becomes $\mathcal{K}'$. The agent also must learn a new set of corresponding action controllers $\mathcal{R}_{\mathcal{K}'}$ (represented as trajectories relative to the arguments of the action). We refer to this scenario as the *novel scenario*. In this novel context, the planner initially uses $\mathcal{K}$ to plan for solving $\mathcal{T}$, once again generating π. Upon executing π, a plan failure occurs for some action $a_f \in \pi$. At this point, the agent must explore its world to learn a new knowledge base $\mathcal{K}'$, providing it with an updated and accurate representation of the new world, along with its corresponding set of action controllers $\mathcal{R}_{\mathcal{K}'}$. We define the learning process $\mathcal{L}$ as the process in which an agent can learn a new knowledge base $\mathcal{K}'$ using exploration method ω, such that $\mathcal{L}(\mathcal{K}, \omega) \mapsto \mathcal{K}'$.

The exploration method ω used by the agent for CPS is a method which can result in knowledge base expansion. For example, in previous work, we demonstrate knowledge base expansion through action discovery (Gizzi et al., 2021a). In preliminary work, we demonstrate knowledge base expansion via action discovery through trajectory segmentation (Gizzi et al., 2019). In another case, we demonstrate action discovery through behavior babbling across action parameter assignments (Gizzi et al., 2021b). In Gizzi et al. (2022), we provide a comprehensive review of work in CPS which provide methods for knowledge base expansion through various exploration methods ω.

Experiments

World Model

We experimented with 2 model types, with each model formulated as multi-label binary classifiers.

Inputs and Outputs The inputs and outputs to our models are listed in Table 1 and Table 2, respectively. Before training our models, we performed basic preprocessing on our data to render the data formats shown in the tables. We one-hot encoded the categorical features in both the input (actions, shapes, color, and orientation were encoded as described in Table 1), and output (world fluent changes were encoded to indicate whether they occurred or not, as described in Table 2). We also standardized continuous features by removing the mean and scaling to unit variance in a given feature space. Lastly, we split our data into a training and testing set to prevent over-fitting.

Model 1: Neural Network The first model we tested was a feed forward neural network (NN), which is a basic artificial neural networks, where connections between nodes do not form a cycle. Our NN had 3 hidden layers, 256 neurons in each hidden layer, a binary cross entropy loss function, and a decaying learning rate for the CPR scenarios. After examining multiple model choices, we determined that a shallow and narrow neural network was not complex enough to learn the data but still achieved high binary accuracy since few actions in the data set affected the agent's world. Conversely, a deep and wide neural network was able to learn the complexity of the data.

Model 2: Naïve Bayes The next model we tested was a naïve bayes model (NB). The NB model uses Bayes Theorem and the assumption that each input variable is independent to dramatically simplify calculations. We extended a binomial naïve bayes model to support multi-label classification by fitting one classifier per label. Recommending actions to the agent in CPS when performing exploration is well suited for a binomial naïve bayes model since the agent is training on a knowledge base of independent trials and each trial produces six binary labels.

Measures We developed four measures used to prioritize exploration trial recommendations by our world models. That is, given a list of possible exploration trials (where each trial describes an action to vary with corresponding parameter settings, and low level information about the entity argument to the action – thus describing a world model input choice), the agent uses its world model to first predict multi-label binary outputs described in Table 2, and then numerically quantifies each trial based on the world model output it render. By using the *least destructive* measure, the model orders the list of recommended exploration trials based on how much a given input changes in it negative reachability output. Exploration trials which minimize these changes are prioritized. The *most changes* measure ranked inputs based on how many fluent property changes they rendered through the world model. Thus, inputs that rendered the highest net value in the sum of the values of Ω were prioritized. The *most positive changes* measure prioritized inputs which resulted in the high rank for the sum of positive reachability, positive touching, and positive visibility outputs. And lastly, the *least negative changes* measure prioritized inputs which resulted in the low rank for the sum of negative reachability, negative touching, and negative visibility outputs.

Scenarios

We ran a proof-of-concept experiment of our methodology in PyBullet, which is a 3D physics simulation environment. The world model of the agent is first trained on input/output data points (described later), sampled from randomized actions on randomized entities. After initial training, the robot

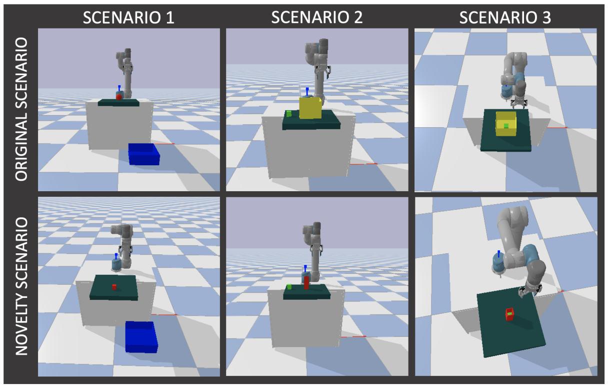


Figure 1: CPS Scenarios. In Scenario 1, the robot has a goal of pushing an object off the table, into a bucket on the ground. In Scenario 2, the robot has a goal of placing an item into a container. In Scenario 3, the robot has a goal of emptying the contents of a container, which has one object in it.

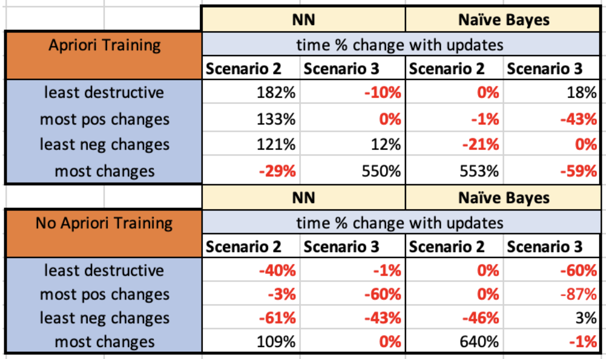
attempts to solve 3 CPS scenarios sequentially Each experimental scenario in shown in terms of its *original* and *novelty* condition, shown in Figure 1).

Results

In order to evaluated whether the use of a world model increases CPS ability (through decreasing exploration time) across longitudinal CPS trials, we ran two experiments, where the world model used in each trial was first trained on the same 200 data points of randomly generated world model interactions. In each experiment, we took the average of four trial runs to calculate average exploration time for each scenario in the described sequence, along with the total exploration time for the sequence. We performed this test for the NN model and the NB model. Additionally, we performed this test for each of the 4 measures.

In the first experiment, we allowed exploration trials during each scenario to be used to train the model over time, across scenarios. In the second experiment, we reset the training data back to the original set of 200 data points, and retrained the model before each scenario. In this way, we were able to observe whether training on CPS trials was helpful toward decreasing CPS exploration over time. Note that each scenario is characteristically different, regarding the amount of exploration needed to find a solution. For example, scenario 1 requires exploration of actions outside of the original failed plan, or defocused exploration. Therefore, comparisons were made relative to the corresponding scenarios of each experiment.

We did not find a significant difference between model updating versus not model updating in the first experiment. We believe this may be due to the fact that the data generated in the randomized trials may have not been a great representation of normal robot exploration (for example, in many trials, objects fell off of the table before exploration was able to begin). Moreover, even with accurate exploration, we believe the training apriori may have biased the agent toward "nominal problem solving," which uses different reasoning than CPS. For this reason, we decided to test how the agent would perform if there was little aprioiri training.



	NN		Naïve Bayes	
Apriori Training	time % change with updates			
	Scenario 2	Scenario 3	Scenario 2	Scenario 3
least destructive	182%	-10%	0%	18%
most pos changes	133%	0%	-1%	-43%
least neg changes	121%	12%	-21%	0%
most changes	-29%	550%	553%	-59%

	NN		Naïve Bayes	
No Apriori Training	time % change with updates			
	Scenario 2	Scenario 3	Scenario 2	Scenario 3
least destructive	-40%	-1%	0%	-60%
most pos changes	-3%	-60%	0%	-87%
least neg changes	-61%	-43%	-46%	3%
most changes	109%	0%	640%	-1%

Figure 2: Percent change in time for scenario 2 and 3 execution. Red values show instances where model updating improve performance (by reducing exploration time against trials with no model updating). Thus, in the case where the world model was first trained a priori, there was a decrease in CPS exploration time in 50% of the measure-model combination choices. In the case where the world model was not trained a priori, there was a decrease in CPS exploration time in 81% of the measure-model combination choices.

We performed the same two experiments, where we instead only trained our models on 4 data points (one for each action, randomly sampled from the original 200 data points). Results are shown in Figure 2. In this case, we found that there was a reduction in CPS time between scenario 1 and 2, *and* between scenario 2 and 3, in 50% and 81% of the trial combinations for NN and NB, respectively (further described in the caption of Figure 2). **This shows that updating the NB world model using only CPS exploration trials is beneficial toward decreasing CPS exploration, as opposed to not.** When executing sequences of consecutive CPS exploration without model updating in-between scenarios, the agent was still updating its own world model within the exploration of an individual scenario. Therefore, its possible that there is still benefit in having a "miniature" world model for each scenario, not to be used in a long term sense.

Conclusion and Future Work

In this paper, we develop a method for enabling life-long creative problem solving by training a world model on creative problem solving exploration data to increase CPS exploration performance. It was shown that using a naive Bayes model is useful toward decreasing exploration time in CPS over time, when trained on CPS data alone. A limitation of our work is that it does not perform CPS over extensive/complex CPS operational runs. Future work should consider performing LLCPS over 100 seeds, or more. Similar limitations are addressed in Sun (2007). Additionally, future work should consider using alternative output vectors for capturing meta-level world changes, and different measures to rank those output values such that predictions can be more consistent across scenarios. Lastly, future work should compare alternative meta-level world models for LLCPS, including reinforcement learning-based methods.

Author Contributions

[EG] ideated/wrote the paper, and developed the framework.
[WWL] co-led development of the simulation environment.
[MGC] co-led development of the simulation environment.
[EH] developed the world models.
[JS] provided advise and edits to the paper.

Acknowledgements

The research presented in this paper was conducted with partial support from DARPA (contract W911NF-20-2-0006).

References

Michael Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020.

Evana Gizzi, Mateo Guaman Castro, and Jivko Sinapov. Creative problem solving by robots using action primitive discovery. In *2019 Joint IEEE 9th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pages 228–233. IEEE, 2019.

Evana Gizzi, Lakshmi Nair, Jivko Sinapov, and Sonia Chernova. From computational creativity to creative problem solving agents. In *International Conference on Computational Creativity (ICCC)*, pages 370–373, 2020.

Evana Gizzi, Mateo Guaman Castro, Wo-Wei Line, and Jivko Sinapov. A framework for creative problem solving through action discovery. In *2021 Robotics: Science and Systems (RSS 2021) Workshop on Declarative and Neurosymbolic Representations in Robot Learning and Control (DNR-ROB)*, 2021a.

Evana Gizzi, Amel Hassan, Wo Wei Lin, Keenan Rhea, and Jivko Sinapov. Toward creative problem solving agents: Action discovery through behavior babbling. In *2021 IEEE International Conference on Development and Learning (ICDL)*, pages 1–7. IEEE, 2021b.

Evana Gizzi, Lakshmi Nair, Sonia Chernova, and Jivko Sinapov. Creative problem solving in artificially intelligent agents: A survey and framework. *arXiv preprint arXiv:2204.10358*, 2022.

Khurram Javed and Martha White. Meta-learning representations for continual learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Gregory Kahn, Pieter Abbeel, and Sergey Levine. Badgr: An autonomous self-supervised learning-based navigation system. *IEEE Robotics and Automation Letters*, 6 (2):1312–1319, 2021.

Dmitry Kalashnikov, Jacob Varley, Yevgen Chebotar, Benjamin Swanson, Rico Jonschkowski, Chelsea Finn, Sergey Levine, and Karol Hausman. Mt-opt: Continuous multi-task robotic reinforcement learning at scale. *arXiv preprint arXiv:2104.08212*, 2021.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113: 54–71, 2019.

Dushyant Rao, Francesco Visin, Andrei Rusu, Razvan Pascanu, Yee Whye Teh, and Raia Hadsell. Continual unsupervised representation learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Ron Sun. The importance of cognitive architectures: An analysis based on clarion. *Journal of Experimental & Theoretical Artificial Intelligence*, 19(2):159–193, 2007.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Metaworld: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*, pages 1094–1100. PMLR, 2020.

Bayesian Modelling of the Well-Made Surprise

Patrick Chieppe, Penny Sweetser and **Eryn Newman**
Australian National University
Canberra ACT 2600 Australia
patrick.chieppe@anu.edu.au, penny.kyburz@anu.edu.au, eryn.newman@anu.edu.au

Abstract

The "well-made" surprise is a narrative pattern of setting up and executing a surprise in a way that is generally perceived as enjoyable and rewarding. It leverages biases in human cognition to manipulate the audience's state of belief, and is commonly found in western culture as early as Aristotle's Poetics. We propose a novel framework to model the audience's beliefs of a narrative world using approximate Bayesian inference over Markov Logic Networks. We operationalise three qualitative attributes of the well-made surprise (consistency, divergence and certainty) as quantitative functions of the outputs of inference. This work follows the paradigm from computational narrative of operationalising qualitative concepts from literary theory in order to model and generate narratives, either autonomously or cooperatively with a human author. We demonstrate the proposed framework on ten short narratives, and test it with a study on 91 participants. We find that for consistency and divergence, a change in the model's prediction corresponds with a significant change in the participants' rating. Our results suggest that the proposed framework may have meaningful predictive power and potential for future applications to narrative generation, plot analysis, and computer-aided creativity.

Introduction

Computational narrative is a long-standing research field focusing on modelling the building blocks of a narrative in a machine-processable structure, most often with the goal of analysing existing narratives or generating novel ones (Kybartas and Bidarra 2017; Valls-Vargas, Zhu, and Ontanon 2017). A common paradigm in generational narrative is to apply some existing algorithmic framework to an operationalisation of a concept from literary theory, such as suspense (Cheong and Young 2008), surprise (Bae and Young 2014) and conflict (Ware 2002). In contrast, other recent examples in the field build upon natural language processing advances using neural networks (Radford et al. 2019) to directly process and generate natural language narratives (Yao et al. 2019; Fan, Lewis, and Dauphin 2018) and have gained public popularity, for instance with the indie game AI Dungeon (Hua and Raley 2020). These rely on the ready availability of large datasets rather than human-encoded models, solving scalability issues but losing transparency and decodability of the model's internal workings in doing so.

Tobin (2018) describes the well-made surprise as a common pattern in western narratives, dating as far back as the well-made tragedy in Aristotle's Poetics, from which they borrow the term. It describes a surprise, or an unexpected event or occurrence in a narrative, that is accompanied by an experience of insight, also called an "Aha" experience (Topolinski and Reber 2010; Skaar and Reber 2020). In this particular scenario, it is the formation of a new understanding of the narrative, associated with suddenness, ease and fluency of processing, certainty in the new understanding and positive affect, which lead to overall enjoyment of the surprise.

Tobin's theory of the well-made surprise deals primarily with literature and film, but it's a construct extensible to any storytelling medium. Thus, for the remainder of this paper we will not assume any particular medium when referring to an author (the person or people crafting the narrative and the surprise), an audience (the person or people experiencing the narrative) and a text (the artifact through which the narrative is conveyed from the author to the audience).

Tobin also details a variety of techniques by which an author may construct a well-made surprise, largely leveraging common biases in human cognition (Evans 1989) to deliberately construct misunderstandings or misinterpretations about the narrative leading up to the surprise, while still enabling the "correct" meaning of the text to be recognised and accepted in retrospect. Tobin argues that it is especially because well-made surprises exploit these biases that they produce an experience of insight. Examples include minimising the audience's attention towards certain information (Emmott and Alexander 2014) and shifting the frame of reference from an objective telling of the events to a character's impression of them using presupposition clauses to mask falsehood as truth (Loftus and Zanni 1975; Bredart and Modolo 1988). Many of these details of the techniques are specific to a medium, but in general, their intended effect is to manipulate how the audience processes information and builds a mental model of the narrative, steering them towards a state in which later events can best deliver a satisfying, insightful surprise.

In this work, we investigate the applicability of Tobin's theory as a modelling tool in the field of computational creativity. We identify three main areas of applications for such a model:

- *Narrative analysis*: Improving the understanding of existing narratives by providing a new analytical lens through which to model surprises (Valls-Vargas, Zhu, and Ontanon 2017).
- *Computer-aided creativity*: Aiding authors in the process of writing a satisfying plot by identifying features such as plot holes and well-made surprises (Kapadia et al. 2015), similarly to how formal methods in software modelling can aid software developers verify their abstractions (Guttag and Horning 1980).
- *Generative narrative evaluation*: Evaluating the output of other narrative generation tools, for example as a search heuristic or as a validation metric. Tobin (2018, pp. 54-55) highlights that the pattern of the well-made surprise is often pleasant to experience even when familiar with it, which is a very attractive property for narrative generation, which often struggles with overcoming repetitiveness (Alabdulkarim, Li, and Peng 2021).

We believe that there is unexplored potential in the computational modelling of the theory of the well-made surprise. There exists significant work in modelling surprise and other related concepts in computational narrative (Bae and Young 2014; Cheong and Young 2008), as well extensive study into the properties of the "Aha" experience in cognitive psychology and neuroscience (Skaar 2019; Skaar and Reber 2020; Chu and MacGregor 2011), and the theory of the well-made surprise points out an important link between surprise and insight. However, no previous work that we are aware of has attempted to bring all of the above together.

Tobin's work bridges the disciplines of narrative theory and cognitive psychology, and in addition doesn't require deep familiarity with either field to understand. We combine this with the approach from computational narrative to operationalise literary theory and cognitive psychology concepts in an effort to bring computer science into the mix and take the first step towards a novel cross-disciplinary computational model of surprise.

We study the mental model that the audience builds of the narrative through the theory of Bayesian probability (Cox 1946; Jaynes, Jaynes, and Bretthorst 2003), focusing on their knowledge or beliefs about the story world and the inferences they perform throughout the narrative (McKoon and Ratcliff 1992; Graesser, Singer, and Trabasso 1994). From such a model, we aim to operationalise key qualities of the well-made surprise by expressing them as functions of the model, based on related ideas from logic and information theory. We implement a probabilistic framework of the well-made surprise and implement three such operationalisations, which we evaluate on a study with 91 participants. We find that the model's predictions agree with participant ratings for two of the three operationalisations, and identify several strengths and weaknesses of the proposed framework.

Background

The Bayesian theory of probability (Cox 1946; Jaynes, Jaynes, and Bretthorst 2003) has seen extensive use in computer science as the theoretical basis for probabilistic models (Russell and Norvig 2010, pp. 510-546), as well as applications in both cognitive sciences (Griffiths, Kemp, and Tenenbaum 2008, pp. 85-138) and literary theory (Kukkonen 2014). Under the Bayesian framework, probabilities represent a degree of belief in a hypothesis, with $P(x) = 1$ representing a certain fact, and $P(x) = 0$ an impossibility. As new data is acquired, existing beliefs are updated using Bayes' theorem. In the context of experiencing a narrative, new beliefs are added to the model as needed in order to make sense of the narrative (McKoon and Ratcliff 1992; Graesser, Singer, and Trabasso 1994), and the resulting model is a combination of information that the narrative has provided and of the audience's own background knowledge.

A Markov Logic Network or MLN (Richardson and Domingos 2006) is a probabilistic model that encodes a joint probability distribution over the truth value of all ground atoms in its domain. Like other Bayesian models it allows for inference, or computing the probability $P(A|B)$ for some A to be true given some known prior B, both being logic formulas. While exact inference is intractable in the general case, efficient approximate inference algorithms have been developed (Riedel 2005; Geier and Biundo 2011; Niu et al. 2012; Van den Broeck 2013).

A MLN is defined as a set of weighted first-order logic statements defined over a finite domain. MLNs afford the expressive power of first-order logic alongside the ability to model uncertainty, both in the sense of information that is varying degrees of plausible rather than absolutely true or false, and in the sense of contradictory information. We find these to be valuable properties in the modelling of the well-made surprise. Partial, uncertain and contradictory information is extremely common in surprising narratives, and the ability to reason about such imperfect information is an important part of understanding well-made surprises, where an initially unexpected outcome is made sense of and obvious in hindsight. In addition, MLNs' expressive power proves especially useful due to the exploratory nature of this work, allowing a wide range of concepts to be modelled.

MLNs can be seen as a template from which a ground Markov Random Field or MRF (Murphy 2012) can be built. The ground MRF is a bipartite graph of all ground atoms and all groundings of all rules, where an edge exists between an atom and a rule if the atom appears in the rule. This interpretation is especially useful for visualising the structure of a MLN and the flow of information during inference, as shown later in this paper.

Literature review

In the field of computational narrative, there are many examples of systems designed to generate stories guided by some operationalised narrative concept. Bae and Young (2014) use AI planning to model flashbacks and foreshadowing in order to construct surprising narratives with explainable causes, and present a methodology to adapt their model to narrative analysis of surprises. Arinbjarnar (2005; 2008) propose an interactive murder mystery plot generation engine based on Bayesian networks which combines ideas from Propp's (1968) morphology of the Russian folktale with accepted genre conventions from mystery writers. Riedl and Bulitko (2012) and Arinbjarnar, Barber, and Kudenko (2009) survey a large body of work on interactive narratives.

Bayesian methods and especially Bayesian networks have seen extensive use in the modelling of uncertainty and knowledge, on both real and fictional narratives and on a very wide variety of topics. These include evidence in legal cases (Vlek et al. 2013), workplace injury narrative coding (Lehto, Marucci-Wellman, and Corns 2009; Measure 2014; Taylor et al. 2014), the visual perception of surprising events while watching television (Itti and Baldi 2009) and how emotion appraisals are transmitted across retellings of a story (Breithaupt, Li, and Kruschke 2022). There are many more examples, see Canaj, Biba, and Kote (2018) for a more thorough survey. Skaar (2019) studies in detail several aspects of the "Aha" experience using Bayesian statistics.

While Markov Logic Networks are less prominent in the literature than Bayesian networks, they have seen several successful applications. Singla and Mooney (2011) train a MLN of a plan from observed actions, Ohwatari et al. (2014) model interpersonal relationships between characters and Patil et al. (2018) use MLNs to identify characters with multiple referring aliases.

Applications to interactive narratives are especially relevant to our research, as the algorithmic infrastructure driving the telling of an interactive narrative can naturally start closer to the world of logic and Bayesian modelling than more traditional media, potentially allowing for a smoother and more direct modelling process. Rowe and Lester (2010) modelled user knowledge in an interactive narrative using dynamic Bayesian networks, while Ha et al. (2012) apply MLN structure learning to their user's goals based on the actions they take in a narrative world.

Qualities of the well-made surprise

Tobin (2018, Chapter 5) identifies several qualities that define the "Aha" experience, and by extension the well-made surprise, which we elaborate on and adapt to our approach in the following sections. Their work focuses on four qualities that are required for an experience of realisation to be the basis of a well-made surprise (suddenness, certainty/confidence, ease/fluency and pleasure/enjoyment). We specify and formalise an additional three which are based on our interpretation of concepts Tobin alludes to throughout their work (coherence, consistency, and divergence).

Coherence

Coherence is a measure of the logical flow in the surprise. An incoherent surprise is unrelated to the rest of the narrative and is confusing even in hindsight. "Cheap" twist endings (Marie-Laure Ryan 2009) often fall into this category, failing to justify their existence in the story world beyond resolving a plot element or dismissing a contradiction (e.g. "it was all a dream", so it doesn't have to make sense).

Consistency

Consistency is the degree to which to which the surprise is compatible with the rest of the story leading up to it. A consistent surprise is plausible given all of the information presented by the story thus far, and the audience is able to integrate it into their understanding of the story world without any unexplainable contradictions emerging. Stories often uphold this by masking contradictions behind a character's subjective impression of events, reframing what originally appeared as factual to be misguided, misunderstood or fabricated.

Divergence

Divergence is the magnitude of the knowledge revision caused by the reveal. A divergent surprise will have deeper, further reaching implications in the plot, and force the audience to revise their understanding of earlier events. This extends the notion of how surprising any single event is (i.e. its probability) with the outcome of the inferences that the new information triggers in the audience.

Suddenness

Suddenness is the speed at which the audience arrives at a new understanding after their previous one is revised. A sudden surprise will cause the audience to revise their understanding and adopt a new one within a short span of time.

Inevitability

Inevitability is the degree to which the final understanding is intuitive, satisfying and (in hindsight) obvious, compared to the initial understanding. This can take on a variety of shapes, such as a character's actions being reframed to be more in line with their motivations, or a previously unimportant detail (a "Chekhov's gun") gaining new meaning. This is closely related to Tobin's ease/fluency concept, but we adopt the term "inevitability" from other parts of their work. We chose this name to focus on the knowledge and reasoning side of the concept (a surprise that can be explained and reasoned about into a likely occurrence in hindsight), rather than the psychological idea of cognitive fluency (the quality of thoughts that are easy to mentally process), although the latter would be an interesting avenue for future research (Oppenheimer 2008).

Certainty

Certainty is the degree to which the new understanding appears as certain and undoubtable, naturally fitting into the story world in such a way that it answers questions and fills gaps in knowledge besides the subject of the surprise.

Enjoyment

When the other qualities hold, we expect the surprise to be enjoyable. Due to the highly subjective nature of the experience, there is a fine line between accepting the surprise as insightful and rejecting it as a cheap writing trick. This becomes more evident the more ambitious the surprise is at unravelling the audience's previous understanding. For

	Sentence	Encoding
1	Katie just had a very long week at work.	$WorkHard(Katie, Weekdays)$
2	*One cannot work hard without working.*	$\neg DoWork(x,t) \rightarrow \neg WorkHard(x,t)$
3	*One is working if they are working hard.*	$WorkHard(x,t) \rightarrow DoWork(x,t)$
4	She couldn't wait for the weekend, she had made plans to relax and watch her favorite tv series.	$WantToWatchShows(Katie, Saturday)$
5	*Being denied a wish can make someone unhappy*	$WantToWatchShows(x,t) \wedge \neg WatchShows(x,t) \rightarrow Unhappy(x,t)$
6	As Saturday morning rolled around, she woke up to a call from her boss.	$Call(Boss, Katie, Saturday)$
7	He asked her if she could come over to work.	$AskToAtWork(Boss, Katie, Saturday)$
8	*One wouldn't go to work on a Saturday unless their boss asked.*	$\neg AskToAtWork(Boss, y, Saturday) \rightarrow \neg AtWork(y, Saturday)$
9	*If one goes to work, it's to do work.*	$AtWork(x,t) \rightarrow DoWork(x,t)$
10	*Katie couldn't work and watch her shows at the same time.*	$\neg (DoWork(Katie,t) \wedge WatchShows(Katie,t))$
11	*One cannot be happy and unhappy.*	$\neg (Happy(x,t) \wedge Unhappy(x,t))$
12	**She happily agreed and had a great time.**	$AtWork(Katie, Saturday) \wedge Happy(Katie, Saturday)$
13	Her boss had noticed how hard everyone worked last week, and threw a party at the office.	$(WorkHard(x, Weekdays) \wedge AtWork(x, Saturday)) \rightarrow Party(x, Saturday)$
14	*One doesn't party and do work.*	$\neg (Party(x,y) \wedge DoWork(x,y))$
15	*Surprise parties make people happy*	$Party(x,t) \rightarrow Happy(x,t)$

Table 1: Example encoding of a story. The reveal is in bold. Background rules are in italics.

instance, surprises relying on unreliable narrators that completely change the perspective of the story from a factual retelling of events to the fallible perception and interpretation of a character can have a polarising effect on their audience.

Proposed model

We view a well-made surprise as composed of three phases: setup, reveal and explanation. During the setup, the audience forms an understanding of the story world, which we call the *flawed* understanding. Then, the reveal is a sudden, surprising event which prompts the audience to question the flawed understanding, and begin forming a new one. The explanation is a final, optional phase in which the story guides the audience towards an improved understanding of the story world, which we call the *truth* understanding.

We model each story as a pair of MLNs, corresponding to the flawed and truth understandings. To demonstrate our modelling process, we wrote ten short stories of four to six sentences each, each story focusing on one of the modelled qualities. For each story, we wrote two variants, one being high in the associated quality, and one low. We wrote the stories such that the difference between the two variants is as minimal as possible to produce the desired difference in the associated quality, while also producing as small a difference as possible in all the other qualities. During the writing process, we categorised stories as high or low in each quality using our subjective judgement.

For each story, we identify one sentence as the reveal, every sentence preceding it as the setup, and every sentence (if any) following it as the explanation. We then encoded each sentence as one or more rules, which are either encoding information stated explicitly in the story, or background knowledge that we assume the audience will draw from in order to make sense of the sentence. Rules and atoms are shared by both the flawed and truth models where possible, as we will later define functions over shared atoms.

See Table 1 for an example encoding of a story written to have high certainty. In the story, Katie is hoping to have a relaxing weekend (4) but is suddenly asked to come to work (7). The audience might expect her to either not abide the request ($\neg AtWork(Katie, Saturday)$), or to begrudgingly do so and be unhappy with the result (due to 5, 9 and 10). The reveal (12) is unexpected because neither holds (due to 11), and is then explained by referring back to the fact that Katie worked hard during the week (1).

We run approximate inference over both the truth and flawed models, using the Alchemy 2 implementation of MLNs (Kok and Domingos 2005) with the default MaxWalkSat and Gibbs sampling approximate inference algorithm described by Richardson and Domingos (2006).

	Atoms
1	$DoWork(Katie, Weekdays)$
2	$WorkHard(Katie, Weekdays)$
3	$AskToGoToWork(Boss, Katie, Saturday)$
4	$WatchShows(Katie, Saturday)$
5	$WantToWatchShows(Katie, Saturday)$
6	$DoWork(Katie, Saturday)$
7	$AtWork(Katie, Saturday)$
8	$Happy(Katie, Saturday)$
9	$Unhappy(Katie, Saturday)$
10	$WorkHard(Katie, Saturday)$
11	$Party(Katie, Saturday)$

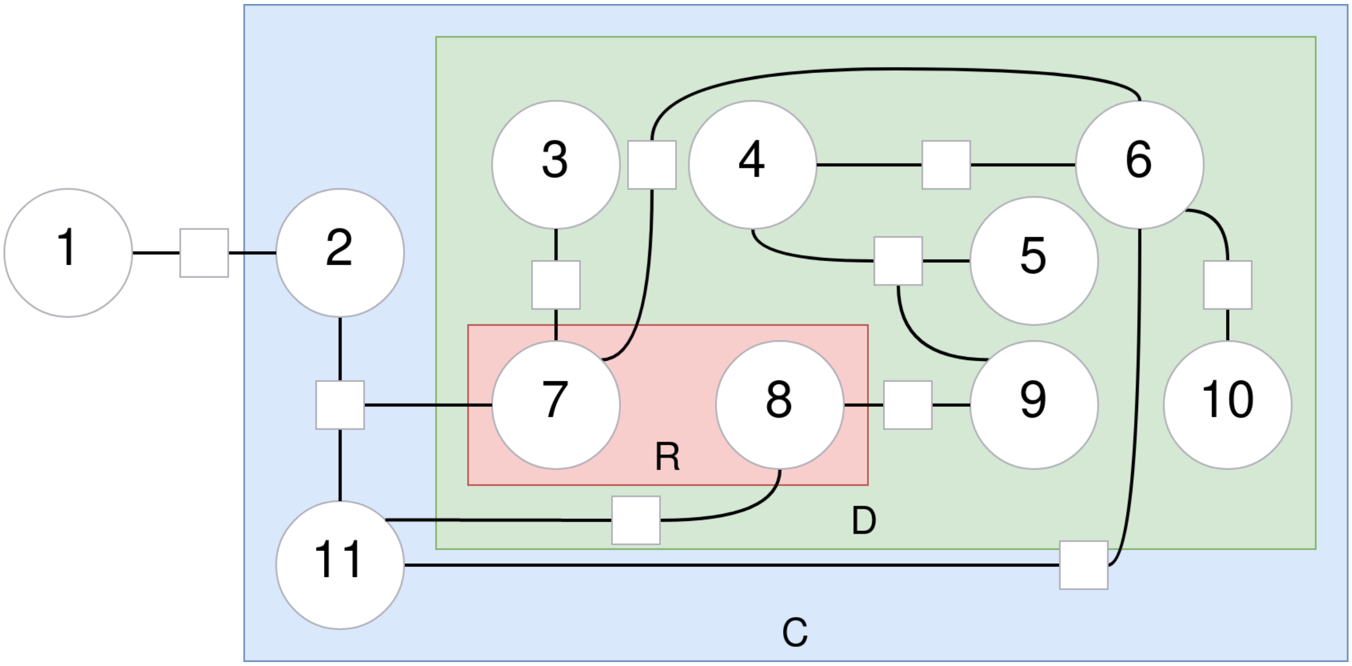


Figure 1: Partial example ground network with reveal (R), divergence blanket (D) and certainty blanket (C) highlighted. Circles are atoms, squares are rules.

This process approximates sampling from the joint probability of all possible worlds defined by the model, and for each ground atom and rule it keeps track of the number of sampled worlds in which they are true. $P(x)$ is then the output probability of x, defined as the proportion of worlds in which x is true among all sampled worlds. We define $P_f(x)$ and $P_t(x)$ to be $P(x)$ in the flawed and truth model respectively.

Operationalisations

From the qualities described earlier in the paper, we operationalise three: consistency, divergence and certainty. Partial work for the operationalisation of coherence was completed, but was not included in the final model. Similarly, inevitability and suddenness are out of the scope of the current model and analysis. Enjoyment involves many subjective factors to the point that it cannot be expressed simply in terms of knowledge and belief, and is not modelled in this work.

Consistency

For the scope of this work, we limit our analysis to instances in which facts directly stated by the narrative contradict each other, and we operationalise this quality as a satisfiability check of the conjunction of all the hard rules in the truth model.

Divergence

We follow an approach similar to Itti and Baldi's (2009) "wow" unit of surprise to quantify the total amount of surprise generated by the reveal.

$$\frac{1}{|D|} \sum_{x \in D} KL(P_t(x) \| P_f(x)) \tag{1}$$

Where KL is the Kullback-Leibler divergence (Kullback and Leibler 1951). We define the divergence blanket D as the set of atoms in common to the flawed and truth ground networks and that are conditionally dependent on the reveal, conditioned on all atoms with known value. In other words, D is the set of all ground atoms that can be reached starting from the reveal, traversing any ground rule edge, and stopping at any atom with known value. D captures the notion of the chain of reasoning that the audience performs to predict the reveal. It aims to capture not only how surprising the reveal is, but also how much this surprise flows backwards through logical links and prompts revision of previously believed information.

Statement	Value
Divergence	
The story is surprising	Positive
The story is not surprising	Negative
The story is not predictable	Positive
The story is predictable	Negative
Consistency	
The story doesn't contradict itself	Positive
The story contradicts itself	Negative
The story made sense	Positive
The story didn't make sense	Negative
Certainty	
The ending is satisfying	Positive
The ending is not satisfying	Negative
The surprise doesn't feel cheap	Positive
The surprise feels cheap	Negative

Table 2: Evaluation statements

Answer	Positive	Negative
Strongly disagree	-1.0	1.0
Somewhat disagree	-0.5	0.5
Neither agree nor disagree	0.0	0.0
Somewhat agree	0.5	-0.5
Strongly agree	1.0	-1.0

Table 3: Likert scale conversion key

It should be noted that $KL(P(x)\|Q(x))$ is not defined when $P(x) = 0$ and $Q(x) \neq 0$, but since we encode hard rules as an arbitrarily large weight rather than an actually infinite weight for computation reasons, output probabilities are never exactly 0 or 1. This has a similar effect to adding a small prior to all probabilities.

Certainty

Shannon entropy (Shannon 1948) is a commonly used measure of uncertainty, and we track its overall change across all modelled information when transitioning from the flawed to the truth model.

$$\frac{1}{|C|}\sum_{x\in C} H(P_f(x)) - H(P_t(x)) \tag{2}$$

Where H is the Shannon entropy. We define the certainty blanket C as the set of atoms in common to the flawed and truth ground networks and that are conditionally dependent on the reveal in either the flawed or truth ground network, conditioned on all atoms with known value. This is defined similarly to D, but note that $C \subseteq D$, as it includes new information that the flawed interpretation had no knowledge of (the audience hadn't thought of it), but that is still relevant to reasoning about the reveal in retrospect. In Figure 1, $\mathit{WorkHard}(\mathit{Katie}, \mathit{Weekdays})$ is not in D since before knowing about the surprise party, Katie's hard work during the week only relates with her desire for a restful weekend. The same atom is in C since it's used in the explanation. $\mathit{DoWork}(\mathit{Katie}, \mathit{Weekdays})$ is in neither, as any rules leading from it to the reveal first go through known atoms.

Evaluation

We evaluate our framework with an exploratory study on a small set of stories, with a total of 91 undergraduate participants recruited through the Australian National University's School of Psychology's Research Participation Scheme.

In this evaluation, we used a fully within-subjects design, focusing on factors that made it into the final framework. We had a 3 (quality: consistency, divergence, certainty) by 2 (variant level: low, high) design. While participants read a total of 10 stories, we focused our analysis on only 7 of them, as 3 stories focused on an operationalisation of coherence that was not included in the final framework. For each quality we operationalised, we used multiple stories to eliminate item specific effects—that is, participants read 3 different stories varying in consistency (low, high). For each of the 7 stories, participants read a version that was high or low on a target quality. For example, for consistency, participants saw a total of 3 stories, in a high and low level of consistency. See supplemental materials for how consistency, divergence and certainty were manipulated as high or low across each story. Each participant was shown all 14 story variants in random order, subject to the restriction that the two variants (low, high) of the same story were always presented one after the other, again in random order. After reading each of the 14 total story versions, participants were asked to evaluate each story across ratings presented in Table 2, as well as rating comprehension of each story. All ratings were answered on a 5-point Likert scale, which were converted according to the key in Table 3 and averaged for each quality. Relatively more positive values as displayed in Table 4 indicate higher ratings of the key dependant variable (e.g. consistency). Answers associated with low comprehension scores (< 0.25) were filtered out as outliers, but the same significant pattern of results is found with those outliers included. Note that in Table 4, we limit our analysis to the key dependant variable of interest—for stories where we varied consistency, we focus our analysis on consistency as per Table 4. Note that other values may be of interest for further analysis, such as interactions between qualities.

Results

For consistency and divergence, a paired-samples two-tailed t-test showed significant change in the mean of participant answers between the two variants of a story ($p < 0.001$), in the same direction as predicted by the model. For certainty, the answers showed less marked change in one of the stories ($p = 0.077$).

Discussion

The results suggest that the framework has meaningful predictive power for the modelled qualities, and in general the approach of using information theoretical functions to model the well-made surprise shows promise.

Title	Variant	Predicted	Mean	Std dev	t-value	p-value
			Consistency			
The Macaroni	Low	0	−0.5443	0.3968	−7.859	< 0.001
	High	1	−0.0016	0.5449		
Sarah's Walk	Low	0	−0.6094	0.4857	−13.762	< 0.001
	High	1	0.5688	0.522		
Catherine at the Beach	Low	0	0.0111	0.5935	−4.902	< 0.001
	High	1	0.3436	0.4554		
			Divergence			
Emma's Move	Low	0.0083	−0.4253	0.3502	−12.389	< 0.001
	High	1.7592	0.3287	0.3804		
Jimmy and the Candy	Low	0.1484	−0.3657	0.3831	−14.297	< 0.001
	High	0.9032	0.4645	0.3101		
			Certainty			
Katie's Weekend	Low	0.0985	−0.3224	0.3669	−6.347	< 0.001
	High	0.1094	−0.0015	0.4052		
Peter Plays Pool	Low	0	−0.2214	0.3241	−1.792	0.077
	High	0.1425	−0.1296	0.3244		

Table 4: Results of model evaluation. Note that the predicted values are not in the same units as the collected data.

The lack of a common unit of measure between model output and collected data makes it difficult to quantify its precision, and a methodology for normalising model outputs to an accepted scale would greatly improve its verifiability.

The result for the last story under certainty ("Peter Plays Pool") may be partially explained by the questions for certainty being very vague statements about the quality of the surprise and of the insight experience, so more specific questions might yield more useful results. This result still highlights how subjective the overall quality of a surprise can be, even for a very short narrative.

Future Work

Future studies should explore further generalisation of the current findings to more general categories of narratives, especially longer narratives and existing corpora of real-world narratives containing well-made surprises. Tobin (2018) touches upon many literary examples throughout their work which future research should strive towards being able to model. The design of future studies should also take into account the ability to generalise across items. Our study's design manipulated each story in an unique way, largely limiting analysis to individual story variation pairs. These questions could also be examined in a between-subjects design, where people do not have the relative comparison across story versions. These are fruitful avenues for future research.

The weakest part of the current framework is matching a model to a narrative. Due to the high flexibility of MLNs, any narrative (even very short ones) can have a wide range of subjective encodings, and two very similar models may produce different outputs. This is a general criticism often raised towards Bayesian modelling in cognitive sciences (Marcus and Davis 2013; Tauber et al. 2017), and is a consequence of the combination of model flexibility, subjective human modelling, and outputs that are sensitive to model formulation. Model consensus procedures such as those used by Trabasso and Sperry (1985) should be used by future research using hand-written models. Another option is to pursue model extraction from questionnaires (Graesser, Robertson, and Anderson 1981). The approach is still inherently not scalable in the context of open narrative generation, and is likely better suited to aid in narrative analysis or as a computer-aided writing tool.

As an alternative to hand-written models, this flexibility also means that MLNs' modelling language subsume many existing structured representations of narratives, and we suggest the development of conversion procedures from existing narrative models to the proposed framework. In particular, the operationalised qualities may find use as heuristics to evaluate the output of generative models which learn their domain representation from existing data (Li et al. 2013) or publicly available corpora (Guan, Wang, and Huang 2019; Swanson and Gordon 2012). Conversely, existing generative frameworks could be adapted to produce narrative variations suitable for use in future studies (Porteous et al. 2010; Piacenza et al. 2011).

It may be possible to extend consistency to a continuous quantity by adapting a MLN weight learning algorithm, such as the voted perceptron (Richardson and Domingos 2006) or the scaled conjugate gradient (Lowd and Domingos 2007). Since MLN weight training is based around computing the optimal weights for each rule given a dataset, we may be able to learn new weights on the samples obtained from inference. Intuitively, conflicting information will cause the respective rules to be false more often than their original weights would imply, and thus result a lower trained weight.

Divergence and certainty are defined over a subset of the marginals, which varies in size depending on model formulation and verbosity. Furthermore, atoms are included in the respective blankets if any rule links to them, with no regard for how important each atom is in the inference process. Future work could draw from research into recall and importance of events (Trabasso and Sperry 1985; Trabasso and van den Broek 1985) to improve them.

The other qualities that haven't been operationalised yet (coherence, suddenness) should also be investigated and modelled in future work. Some, like inevitability, may benefit from being further decomposed into constituent parts in order to be more easily modelled.

Conclusions

We presented a novel cross-disciplinary modelling framework for the well-made surprise. The proposed framework takes the first step in a cross-disciplinary effort to bring the literary theory of the well-made surprise into the world of computer science, drawing from the field of cognitive science along the way to inform the design of the models and research direction. We believe the framework to have potential in the field of computational narrative and creativity, and identify three main areas of promising application as narrative analysis, computer-aided creativity and generative narrative evaluation. We supported our claims with a pilot study, and examined ways in which the framework may be improved and further developed.

Author Contributions

Patrick Chieppe was responsible for writing the manuscript and all other research work not otherwise attributed to the other authors below.

Penny Sweetser advised on the overall course of research and writing, provided frequent feedback and considerably helped shape the direction of this work.

Eryn Newman contributed to the evaluation design and provided initial feedback on the manuscript.

Acknowledgements

This research is supported by an Australian Government Research Training Program (RTP) Scholarship.

Supplementary material

Supplementary material including the stories, models, survey and dataset are available at:

`https://github.com/Palladinium/iccc22`

References

Alabdulkarim, A.; Li, S.; and Peng, X. 2021. Automatic Story Generation: Challenges and Attempts. In *Proceedings of the Third Workshop on Narrative Understanding*, 72–83. Stroudsburg, PA, USA: Association for Computational Linguistics.

Arinbjarnar, M.; Barber, H.; and Kudenko, D. 2009. A critical review of interactive drama systems. In *AISB 2009 Symposium. AI and Games*.

Arinbjarnar, M. 2005. *Murder She Programmed: Dynamic Plot Generating Engine for Murder Mystery Games*. Ph.D. Dissertation, Reykjavík University.

Arinbjarnar, M. 2008. Dynamic Plot Generating Engine. *Proceedings of the Workshop on Integrating Technologies for Interactive Stories (INTETAIN 2008)*.

Bae, B. C., and Young, R. M. 2014. A computational model of narrative generation for surprise arousal. *IEEE Transactions on Computational Intelligence and AI in Games* 6(2):131–143.

Bredart, S., and Modolo, K. 1988. Moses strikes again: Focalization effect on a semantic illusion. *Acta Psychologica* 67(2):135–144.

Breithaupt, F.; Li, B.; and Kruschke, J. K. 2022. Serial reproduction of narratives preserves emotional appraisals. *Cognition and Emotion* 1–21.

Canaj, E.; Biba, M.; and Kote, N. 2018. Bayesian Networks: A State-Of-The-Art Survey. *CEUR Workshop Proceedings* 2280:31–40.

Cheong, Y. G., and Young, R. M. 2008. Narrative generation for suspense: Modeling and evaluation. *Lecture Notes in Computer Science* 5334 LNCS:144–155.

Chu, Y., and MacGregor, J. N. 2011. Human Performance on Insight Problem Solving: A Review. *The Journal of Problem Solving* 3(2):119–150.

Cox, R. T. 1946. Probability, Frequency and Reasonable Expectation. *American Journal of Physics* 14(1):1–13.

Emmott, C., and Alexander, M. 2014. Foregrounding, burying and plot construction. In Stockwell, P., and Whiteley, S., eds., *The Cambridge Handbook of Stylistics*. Cambridge: Cambridge University Press. 329–343.

Evans, J. S. B. T. 1989. *Bias in human reasoning: Causes and consequences.* Essays in cognitive psychology. Hillsdale, NJ, US: Lawrence Erlbaum Associates, Inc.

Fan, A.; Lewis, M.; and Dauphin, Y. 2018. Hierarchical Neural Story Generation. *arXiv preprint arXiv:1805.048331*.

Geier, T., and Biundo, S. 2011. Approximate online inference for dynamic Markov logic networks. *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI* 764–768.

Graesser, A. C.; Robertson, S. P.; and Anderson, P. A. 1981. Incorporating inferences in narrative representations: A study of how and why. *Cognitive Psychology* 13(1):1–26.

Graesser, A. C.; Singer, M.; and Trabasso, T. 1994. Constructing inferences during narrative text comprehension. *Psychological Review* 101(3):371–395.

Griffiths, T. L.; Kemp, C.; and Tenenbaum, J. B. 2008. Bayesian models of cognition. *Wiley Interdisciplinary Reviews: Cognitive Science* 1(6):811–823.

Guan, J.; Wang, Y.; and Huang, M. 2019. Story ending generation with incremental encoding and commonsense knowledge. *Proceedings of the AAAI Conference on Artificial Intelligence1* 33(1):6473–6480.

Guttag, J., and Horning, J. J. 1980. Formal specification as a design tool. In *Proceedings of the 7th ACM SIGPLAN-SIGACT symposium on Principles of programming languages - POPL '80*, 251–261. New York, New York, USA: ACM Press.

Ha, E. Y.; Rowe, J. P.; Mott, B. W.; and Lester, J. C. 2012. Goal Recognition with Markov Logic Networks for Player-Adaptive Games. *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence* 2113–2119.

Hua, M., and Raley, R. 2020. Playing with unicorns: AI dungeon and citizen NLP. *Digital Humanities Quarterly* 14(4):1–27.

Itti, L., and Baldi, P. 2009. Bayesian surprise attracts human attention. *Vision Research* 49(10):1295–1306.

Jaynes, E. T.; Jaynes, E. T. J.; and Bretthorst, G. L. 2003. *Probability Theory: The Logic of Science*. Cambridge University Press.

Kapadia, M.; Falk, J.; Zünd, F.; Marti, M.; Sumner, R. W.; and Gross, M. 2015. Computer-assisted authoring of interactive narratives. *Proceedings of the 19th Symposium on Interactive 3D Graphics and Games, i3D 2015* 85–92.

Kok, S., and Domingos, P. 2005. Learning the Structure of Markov Logic Networks. In *Proceedings of the 22nd International Conference on Machine Learning*, 441–448.

Kukkonen, K. 2014. Bayesian narrative: Probability, plot and the shape of the fictional world. *Anglia* 132(4):720–739.

Kullback, S., and Leibler, R. A. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics* 22(1):79–86.

Kybartas, B., and Bidarra, R. 2017. A Survey on Story Generation Techniques for Authoring Computational Narratives. *IEEE Transactions on Computational Intelligence and AI in Games* 9(3):239–253.

Lehto, M.; Marucci-Wellman, H.; and Corns, H. 2009. Bayesian methods: A useful tool for classifying injury narratives into cause groups. *Injury Prevention* 15(4):259–265.

Li, B.; Lee-Urban, S.; Johnston, G.; and Riedl, M. 2013. Story generation with crowdsourced plot graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 27, 598–604.

Loftus, E. F., and Zanni, G. 1975. Eyewitness testimony: The influence of the wording of a question. *Bulletin of the Psychonomic Society* 5(1):86–88.

Lowd, D., and Domingos, P. 2007. Efficient weight learning for Markov logic networks. *Proceedings of the Eleventh European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2007)* 200–211.

Marcus, G. F., and Davis, E. 2013. How Robust Are Probabilistic Models of Higher-Level Cognition? *Psychological Science* 24(12):2351–2360.

Marie-Laure Ryan. 2009. Cheap Plot Tricks, Plot Holes, and Narrative Design. *Narrative* 17(1):56–75.

McKoon, G., and Ratcliff, R. 1992. Inference during reading. *Psychological Review* 99(3):440–466.

Measure, A. 2014. Automated Coding of Worker Injury Narratives. *Joint Statistical Meetings* 2124–2133.

Murphy, K. P. 2012. Undirected Graphical Models (Markov Random Fields). In *Machine Learning: A Probabilistic Perspective*. MIT Press. chapter 19, 661–705.

Niu, F.; Zhang, C.; Ré, C.; and Shavlik, J. 2012. Scaling inference for Markov logic via dual decomposition. *Proceedings - IEEE International Conference on Data Mining, ICDM* (1):1032–1037.

Ohwatari, Y.; Kawamura, T.; Sei, Y.; Tahara, Y.; and Ohsuga, A. 2014. Estimation of character diagram from open movie database using Markov logic network. In *CEUR Workshop Proceedings*, volume 1312, 124–127.

Oppenheimer, D. M. 2008. The secret life of fluency. *Trends in Cognitive Sciences* 12(6):237–241.

Patil, S.; Pawar, S.; Hingmire, S.; Palshikar, G.; Varma, V.; and Bhattacharyya, P. 2018. Identification of Alias Links among Participants in Narratives. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 63–68. Stroudsburg, PA, USA: Association for Computational Linguistics.

Piacenza, A.; Guerrini, F.; Adami, N.; Leonardi, R.; Teutenberg, J.; Porteous, J.; and Cavazza, M. 2011. Generating story variants with constrained video recombination. *MM'11 - Proceedings of the 2011 ACM Multimedia Conference and Co-Located Workshops* 223–232.

Porteous, J.; Benini, S.; Canini, L.; Charles, F.; Cavazza, M.; and Leonardi, R. 2010. Interactive storytelling via video content recombination. *MM'10 - Proceedings of the ACM Multimedia 2010 International Conference* 1715–1718.

Propp, V. I. 1968. *Morphology of the Folktale*, volume 9. University of Texas Press.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; and Others. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1.

Richardson, M., and Domingos, P. 2006. Markov logic networks. *Machine Learning* 62(1-2 SPEC. ISS.):107–136.

Riedel, S. 2005. Improving the Accuracy and Efficiency of MAP Inference for Markov Logic. *Network* 468–475.

Riedl, M. O., and Bulitko, V. 2012. Interactive Narrative: An Intelligent Systems Approach. *AI Magazine* 34(1):67.

Rowe, J. P., and Lester, J. C. 2010. Modeling User Knowledge with Dynamic Bayesian Networks in Interactive Narrative Environments. *Proc. 6th Annu. AI Interact. Digital Entertain. Conf.* 57–62.

Russell, S. J., and Norvig, P. 2010. *Artificial intelligence: a modern approach*. Upper Saddle River, N.J: Prentice Hall, 3rd ed. edition.

Shannon, C. E. 1948. A Mathematical Theory of Communication. *Bell System Technical Journal* 27(3):379–423.

Singla, P., and Mooney, R. J. 2011. Abductive markov logic for plan recognition. *Proceedings of the National Conference on Artificial Intelligence* 2:1069–1075.

Skaar, Ø. O., and Reber, R. 2020. The phenomenology of aha-experiences. *Motivation Science* 6(1):49–60.

Skaar, Ø. O. 2019. *Moments of Brilliance: Understanding the Aha-experience through Bayesian Statistics*. Ph.D. Dissertation, University of Oslo.

Swanson, R., and Gordon, A. S. 2012. Say anything: Using textual case-based reasoning to enable open-domain interactive storytelling. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 2(3):1–35.

Tauber, S.; Navarro, D. J.; Perfors, A.; and Steyvers, M. 2017. Bayesian Models of Cognition Revisited: Setting Optimality Aside and Letting Data Drive Psychological Theory. *Psychological review* 124(4):410–441.

Taylor, J. A.; Lacovara, A. V.; Smith, G. S.; Pandian, R.; and Lehto, M. 2014. Near-miss narratives from the fire service: A Bayesian analysis. *Accident Analysis and Prevention* 62(2014):119–129.

Tobin, V. 2018. *Elements of Surprise: Our Mental Limits and the Satisfactions of Plot*. Harvard University Press.

Topolinski, S., and Reber, R. 2010. Gaining insight into the "Aha" experience. *Current Directions in Psychological Science* 19(6):402–405.

Trabasso, T., and Sperry, L. 1985. Causal Relatedness and the Importance of Narrative Events. *Journal of Memory and Language* 24(1894):595–611.

Trabasso, T., and van den Broek, P. 1985. Causal thinking and the representation of narrative events. *Journal of Memory and Language* 24(5):612–630.

Valls-Vargas, J.; Zhu, J.; and Ontanon, S. 2017. From computational narrative analysis to generation: A preliminary review. *ACM International Conference Proceeding Series* Part F1301.

Van den Broeck, G. 2013. *Lifted Inference and Learning in Statistical Relational Models*. Ph.D. Dissertation, KU Leuven.

Vlek, C.; Prakken, H.; Renooij, S.; and Verheij, B. 2013. Modeling crime scenarios in a Bayesian network. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law - ICAIL '13*, 150. New York, New York, USA: ACM Press.

Ware, S. G. 2002. A Plan-Based Model of Conflict for Narrative Reasoning and Generation. 52(1):1–5.

Yao, L.; Peng, N.; Weischedel, R.; Knight, K.; Zhao, D.; and Yan, R. 2019. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 7378–7385.

Ranking Creative Language Characteristics in Small Data Scenarios

Julia Siekiera[1], Marius Köppel[2], Edwin Simpson[3,4], Kevin Stowe[4], Iryna Gurevych[4], Stefan Kramer[1]

[1]Dept. of Computer Science and [2]Institute for Nuclear Physics, Johannes Gutenberg-Universität Mainz,
{siekiera,mkoeppel}@uni-mainz.de, kramer@informatik.uni-mainz.de,
[3]Dept. of Computer Science, University of Bristol, edwin.simpson@bris.ac.uk,
[4]Ubiquitous Knowledge Processing Lab, Technische Universität Darmstadt
https://www.informatik.tu-darmstadt.de/ukp

Abstract

The ability to rank creative natural language provides an important general tool for downstream language understanding and generation. However, current deep ranking models require substantial amounts of labeled data that are difficult and expensive to obtain for new domains, languages and creative characteristics. A recent neural approach, DirectRanker, reduces the amount of training data needed but has not previously been used to rank creative text. We therefore adapt DirectRanker to provide a new deep model for ranking creative language with small numbers of training instances, and compare it with a Bayesian approach, Gaussian process preference learning (GPPL), which was previously shown to work well with sparse data. Our experiments with short creative language texts show the effectiveness of DirectRanker even with small training datasets. Combining DirectRanker with GPPL outperforms the previous state of the art on humor and metaphor novelty tasks, increasing Spearman's ρ by 25% and 29% on average. Furthermore, we provide a possible application to validate jokes in the process of creativity generation.

Introduction

To process or evaluate creative language, natural language processing systems need to recognise figurative and humorous expressions, so that they do not interpret jokes or metaphors literally, and can gauge different aspects of creativity. The simple binary recognition of figurative or humorous language is not sufficient, as different examples require varying degrees of creativity, and hence different kinds of processing. Consider the following metaphors:

- This view has been **attacked** on the grounds that it...
- She **attacked** the sandwiches like a starving bear.

In both examples, the verb 'attack' strays from the literal meaning of a military offensive, but the first usage is very conventional, while the second appears much more novel and metaphoric. Other properties of creative language, such as humor, have similar gradations, which motivates methods for ranking sentences according to these properties.

The process of creativity is highly complex and nontrivial to automate, but creative writers may benefit from automated tools. As the comedy writer Charlie Skelton said: *to begin with, we must ask: what is the metric for a "successful" joke? ...Is it one that makes the most people laugh, or the right people laugh, or its own creator laugh?* (Skelton 2021), the success of a joke is strongly cultural-based. He also describes the creative process of a professional comedy writer creating a joke: *a joke can be judged, and just as many checkboxes to tick on its journey from the writer's mind to the audience's ears.* In the setup of the Componential Model of Creativity (Press 2017; Press 2011) this can be seen as the response validation and communication step. To help with this step, a ranking model could provide an automated evaluation method to help a comedy writer answer the question: "Is this joke the one that makes the most people laugh?". Ranking models trained with data annotated from various cultural backgrounds could also give insights into how they may perceive different jokes.

To obtain training data for a ranking model, annotators could assign scores to individual examples, but inconsistencies can arise between annotators and across the labels of a single annotator over time. We therefore turn to pairwise comparisons between examples, which simplify the annotators' task and avoid the need to calibrate their scores. A ranker can then derive the entire ranking from pairwise labels. Considering the cost of annotating data for different domains, languages and aspects of creativity, we need a ranker that can be trained on datasets with a small number of examples and sparse pairwise labels. For ranking creative language, Simpson and others (2019) adopted *Gaussian process preference learning (GPPL)*, a Bayesian approach that uses word embeddings and linguistic features and can cope with sparse and noisy pairwise labels. However, it is a shallow model that relies on predetermined features to represent each example.

In contrast, neural network architectures can learn representations directly from pairwise comparisons, but demand a higher quantity of training labels. A recent method, *DirectRanker* (Köppel and others 2019) improves label efficiency for document ranking by fulfilling the requirements of a total quasiorder in the model architecture, which results in faster convergence than other neural network ranking approaches, as this order does not have to be learned. This paper adapts DirectRanker to text ranking for the first time, setting a new state of the art for humor and metaphor novelty, showing that even with limited data, text ranking can benefit from deep representation learning. Our experiments show that combining Bayesian and neural approaches using stacking can improve further ranking quality. While we find a clear benefit to BERT embeddings (Devlin and

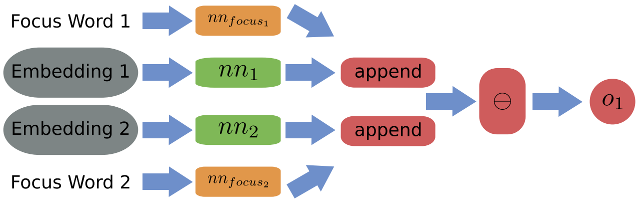


Figure 1: The adapted DirectRanker architecture. Embeddings are fed into the parameter sharing networks nn_1 and nn_2 to generate representations (feature part). For datasets containing focus word information, we add parameter sharing networks nn_{focus_1} and nn_{focus_2}. The (appended) representations are subtracted and fed into the ranking part (in red) with output neuron o_1 that has no bias and $\tanh$ as activation.

others 2018) for humor, current embedding methods have difficulty in modelling metaphors. To support the evaluation of creative content, we make our software available at `https://zenodo.org/record/6275546`.

Related Work

Algorithms solving the ranking problem can be divided into three categories. *Pointwise* rankers assign a score to each document (Cooper and others 1992). *Pairwise* models predict which document is more relevant out of two for a given query (Köppel and others 2019). *Listwise* algorithms optimise a loss function that considers the whole ordered list of documents (Cao and others 2007). Previous research on document ranking combined BERT (Devlin and others 2018) with different learning-to-rank methods of all three categories. While Han and others (2020) and Qiao and others (2019) embed concatenated queries and documents with BERT and fine-tune ranking performance using an arbitrary artificial neural network ranker, Nogueira and others (2019) introduce a multi stage pipeline containing a pointwise and a pairwise BERT ranker to trade off ranking quality against latency. However, these approaches are evaluated neither for small training data scenarios nor on the difficult task of creative language and lack the label-efficient learning property that DirectRanker introduces. In the past, DirectRanker was used for ranking multilingual BERT models (Chen and Ritter 2020), but the approach ranks the models themselves rather than text documents, which we address here.

DirectRanker for Text Ranking

DirectRanker, shown in Figure 1, consists of a *feature* part, which learns a low-dimensional latent representation of the input documents, and a *ranking* part, which receives the latent representations for a pair of examples and predicts a pairwise label. The ranking part is used to train the model from pairwise labels, but can also be used after training to predict the degree of creativity for any arbitrary text.

To adjust the DirectRanker to text ranking, we include dropout layers and batch normalization in the networks nn_1 and nn_2 to reduce overfitting. For some creative language tasks such as metaphor novelty prediction, the aim is to evaluate the use of a specific word or phrase within a larger context. Hence we need to represent both the word or phrase (henceforth the *focus word*) and the sentence that contains it. During initial experiments, we found that transforming the sentence and focus word together in nn_1 and nn_2 leads to unequal weighting of both information sources in the feature part, as the two feature vectors differ in length and in their most extreme values. We therefore add the networks nn_{focus_1} and nn_{focus_2} to the feature part to process the focus words separately from their context. This facilitates training as the model is able to weight the compressed sentence and focus word information in the less complex ranking part. The results of both the sentence network and the focus word network are concatenated and passed to the ranking part.

The ranking function is given by $o_1(x_1, x_2) = \tau\left(w\left(\frac{(u_1,u_{f_1})-(u_2,u_{f_2})}{2}\right)\right)$, where $u_1 = nn_1(x_1)$ and $u_2 = nn_2(x_2)$ compress the input feature vectors x_1 and x_2 to latent representations u_1 and u_2, u_{f_1} and u_{f_2} are latent representations for the focus words computed by nn_{focus_1} and nn_{focus_2}, w represents the multilayer perceptron ranking weights for the last neuron and τ is an antisymmetric sign conserving activation. The loss function remains the same as in the original DirectRanker paper: $L_{\text{rank}}(\Delta y, x_1, x_2) = (\Delta y - o_1(x_1, x_2))^2$, where Δy is the gold pairwise label in the training set. Beside the changes of the feature part, we included the possibility to change the ranking part to a Gaussian Process layer using a Matérn kernel, enabling a direct combination with the ideas of the GPPL model. Therefore, the original ranking function can be replaced with $p(x_1 \succ x_2) = \Phi\left(\frac{u_1 - u_2}{\sqrt{2\sigma^2}}\right)$ for the ranking part, where $x_1 \succ x_2$ indicates that instance x_1 was labeled as preferred to x_2, Φ is the probit function, and σ^2 is a variance parameter.

Text Representation We investigate three text representations. First we choose mean *word2vec* embeddings (MWE) trained on part of Google News (Mikolov and others 2013) to directly compare the findings of Simpson and others (2019) with the DirectRanker. However, *word2vec* embeddings have the disadvantage that they assign a single, fixed representation for each word, even though it may take on different meanings in different contexts, particularly with regard to creative language. To address this, we fine-tune BERT with DirectRanker to produced contextual word embeddings, and again take the mean to represent the whole sentence. To better capture the meaning of a whole sentence, we apply sentence transformers (Reimers and Gurevych 2019) to generate sentence embeddings (SEs). In contrast to MWEs, sentence transformers learn how to compose individual contextual word embeddings and assign sentences with similar meanings close representations in the vector space.

Datasets

We explore GPPL and DirectRanker on two datasets including different types of creative language. The *humor* dataset (Simpson and others 2019) is an extension of Miller and others (2017), which contains 4030 samples with various degrees of humorousness, with an average sentence length of 11 words. The humorous examples can be grouped into

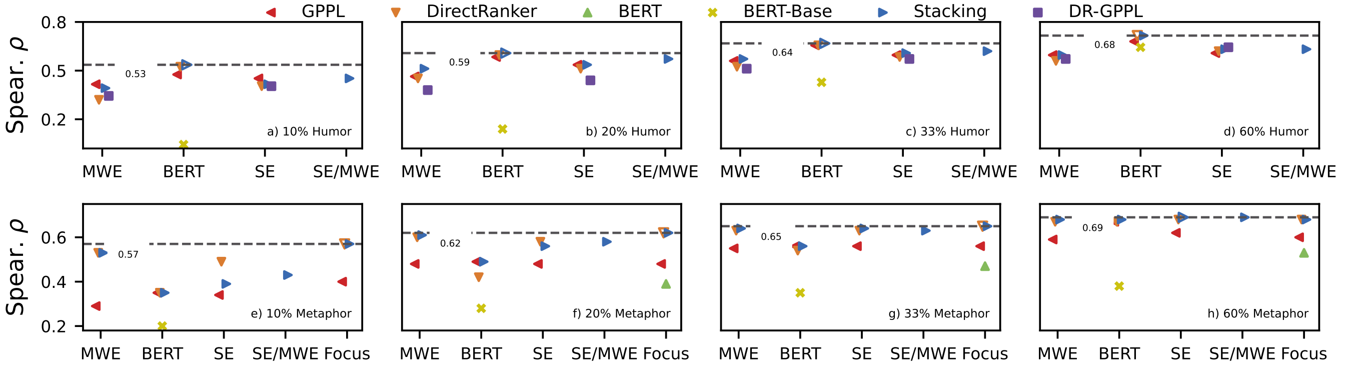


Figure 2: Mean results with different training set sizes. Humor results are shown in the top row, Metaphor results in the bottom row. Each plot shows different splits of the datasets. The different embeddings used by the models are marked on the x-axis. The Stacking method has the possibility to use both SE and MWE. We show Spearman's ρ against the gold score. For better visibility we excluded the result for BERT with focus word embeddings for 10% Metaphor (Spearman's ρ of -0.03) and we added different x-axis offset for the models. A detailed table of the displayed data can be found in Table 1.

homographic and heterographic puns containing purely verbal humor while the non-humorous section contains proverbs and aphorisms. The *metaphor* dataset (Do Dinh and others 2018) contains 72816 examples from the VU Amsterdam Metaphor Corpus (Steen and others 2010) that have been annotated for metaphor novelty, including metaphors in four genres: news, fiction, conversation transcripts, and academic texts. Each example consists of a sentence and a labeled focus word. Both datasets were labeled using crowdsourcing. For the humor dataset, every instance was selected for 14 random pairwise comparisons and each pair was labeled by 5 different annotators. For the metaphor dataset, each instance was included in 6 random tuples, each containing 4 instances. Each tuple was labeled by 3 annotators, who selected the most novel and most conventionalised examples from each tuple. We did not survey the background of the annotators, other than to check that they are proficient in English. We generate pairwise labels between the most novel and most conventionalised samples in each tuple, following Simpson and others (2019). The resulting pairwise comparisons are labeled 1.55 times on average and each instance is present in 8.6 pairs on average.

Experimental Setup

We evaluate our experiments using 4 internal folds for finding the best hyperparameters and 3 external folds for evaluation. To examine the ranking performance on sparse data, we also experiment with artificially reducing training set sizes. For this purpose, we randomly select 60%, 33%, 20% and 10% of the example IDs and train on only the pairs where both examples are in our selection. The remaining samples are used in the test set to counteract the model variation for smaller training sets. The DirectRanker feature part is a 4-layer fully-connected network with 2k, 500, 64 and 7 neurons in each layer. To evaluate the effect of the Gaussian Process layer in the ranking part, we run the experiments on the humor dataset two times, once with and once without the Gaussian Process layer. Code from Simpson and others (2019) was used to train and obtain predictions from GPPL using a Matérn $\frac{3}{2}$ kernel. To improve the overall ranking performance, we combine the predictions of GPPL and the DirectRanker with stacking, using a linear regression model to weight the predictions of the two models. To generate SEs, we use the pretrained 'bert-base-nli-stsb-mean-tokens' model. We use 'bert-base-cased' for fine-tuning BERT with the DirectRanker and reuse the resulting embeddings with GPPL. The methods are evaluated by computing the linear rank correlation between the prediction and the gold standard with Spearman's ρ.

Results of Method Comparison

The results are shown in Figure 2. As a baseline, we include BERT regression models fine-tuned directly on the rankings in the training sets (indicated in Figure 2 with a gold x as BERT-Base). For the metaphor data, we extend the BERT regression model to incorporate the *word2vec* embedding of the focus word as a further input (indicated in Figure 2 with the green $\triangle$). In all cases, both BERT regression and the state-of-the-art GPPL are out-performed by either DirectRanker and Stacking. We highlighted the best model by adding a horizontal line annotated with the Spearman's ρ value and removing the filling. The standard deviation ranges from 0.016 for 60% to 0.038 for 10% on Humor and from 0.006 for 60% to 0.043 for 10% on Metaphor dataset.

On the humor dataset, the BERT baseline performs well in the 60% case as it is able to classify the less relevant documents better. However, the baseline is not suitable for scenarios with less data, in which the pairwise models achieve significantly better results. On Humor, GPPL outperforms the DirectRanker on almost all training set sizes and text representations except for BERT and 60%. The 60% case with SE was the only one where the Gaussian Process layer in the ranking part (DR-GPPL) outperforms the normal DirectRanker approach. Both GPPL and the DirectRanker benefit most from BERT, but the DirectRanker particularly benefits from the pretrained BERT with small training sets. By combining GPPL and DirectRanker, both with BERT, stack-

	Humor				Metaphor			
	60%	33%	20%	10%	60%	33%	20%	10%
Bert Baseline	0.62	0.44	0.20	0.12	0.38	0.35	0.28	0.20
Bert + Focus Word	-				0.53	0.47	0.39	-0.03
GPPL MWE	0.54	0.53	0.47	0.41	0.58	0.55	0.51	0.35
DirectRanker MWE	0.54	0.50	0.44	0.30	0.64	0.60	0.52	0.37
DR-GPPL SE	0.62	0.56	0.45	0.42	-			
DR-GPPL MWE	0.56	0.51	0.40	0.37	-			
Stacking MWE/MWE	0.58	0.56	0.51	0.41	0.68	0.64	0.61	0.53
Stacking BERT/BERT	0.68	0.64	0.59	0.53	0.68	0.56	0.49	0.35
Stacking SE/SE	0.61	0.59	0.53	0.43	0.69	0.64	0.56	0.39
Stacking SE/MWE	0.61	0.60	0.56	0.46	0.69	0.63	0.58	0.43
GPPL ∅ MWE	0.58	0.55	0.47	0.43	0.59	0.55	0.48	0.29
GPPL ∅ BERT	0.65	0.63	0.57	0.48	0.67	0.56	0.49	0.35
GPPL ∅ SE	0.59	0.58	0.53	0.46	0.62	0.56	0.48	0.34
DirectRanker ∅ MWE	0.55	0.52	0.46	0.35	0.67	0.63	0.60	0.53
DirectRanker ∅ BERT	0.68	0.63	0.58	0.52	0.67	0.54	0.42	0.35
DirectRanker ∅ SE	0.62	0.57	0.51	0.42	0.68	0.63	0.58	0.49
Stacking Focus Word	-				0.68	0.65	0.62	0.57
GPPL ∅ Focus Word	-				0.60	0.56	0.48	0.40
DirectRanker ∅ Focus Word	-				0.68	0.65	0.62	0.57

Table 1: Mean results with different training set sizes on the two datasets. We show Spearman's ρ against the gold score. The ∅ indicates that the model's mean score of the 4-fold cross-validation ensemble is evaluated (see the end of Section Stacking. For stacking we first name the embeddings used for GPPL and then for DirectRanker.

ing is able to improve the individual performances across all training set sizes. A similar improvement is shown for other stacking setups, for example with GPPL on SEs and DirectRanker on MWEs (Stacking SE/MWE).

On the metaphor dataset the models' behavior changes. The BERT baseline is not able to reach competitive results in any training scenario and the BERT embeddings do not consistently improve over other embeddings, supporting previous results where BERT underperforms on metaphor tasks (Mao, Lin, and Guerin 2019). DirectRanker outperforms GPPL on most combinations, especially on smaller training sets. For instance, DirectRanker outperforms GPPL with MWE and SE, including for 10% and 20% datasets, showing its suitability for small datasets. In most settings, stacking maintains or slightly exceeds the ranking performance in each combination. In the 20% and 10% case, stacking falls below the maximum individual performance on the SEs as GPPL overfits on the validation set. This might be an effect of learning with SEs on a small training and validation set so that they are not representative of the test set. For metaphor novelty, the models trained on only the *word2vec* focus word embedding outperform those that are also trained with sentence representations with 33% - 10% training data. Furthermore, neither GPPL nor DirectRanker are able to extract much useful information from the sentences alone. With SEs in the 60% case, DirectRanker and GPPL reach a Spearman's ρ of 0.64 and 0.58, respectively. While this may reflect a limitation of the sentence representations, it is also possible that the annotators who produced the gold standard fixated too strongly on the focus words.

Conclusion

In this work we investigated a pairwise ranking approach for creative language based on adapting a recent neural architecture, DirectRanker, that can learn efficiently from small training sets. We combined it with a Bayesian model, GPPL, and evaluated the behavior of all models on the tasks of predicting humorousness and metaphor novelty with different text representations. Despite the expectation that neural networks suffer from overfitting on small datasets, DirectRanker was able keep up with or even improve on GPPL. The proposed stacking approach clearly outperforms state-of-the-art results and is a powerful tool for language tasks with a limited number of training documents. On the humor dataset we showed a substantial ranking improvement over pretrained embeddings by fine-tuning BERT with the DirectRanker architecture. Due to the heavy reliance on the focus word information, this was less effective for the metaphor dataset, where the best results were achieved using only the focus words' *word2vec* embeddings. Resent work showed that using the integration of constructional semantics and conceptual metaphor showed better generalizations across metaphoric and non-metaphoric language (Sullivan 2016). While others provided alternatives to the representation of contextual information, such as the cultural context (Cabezas-García and Reimerink 2022). Using these different approaches could be beneficial for providing better representations for metaphor.

A possible application, in the context of joke generation, is the evaluation of creative content. The ranked sentences can help to evaluate jokes and quantify whether they are funny for a majority of people. Nevertheless, this method comes with limitations since it is not aware of any context the joke was made. To see this, consider this cherry-picked example of a joke which was ranked high: "I do a lot of **spreadsheets** in the office so you can say I'm **excelling** at work." While the model was able to characterize that this sentence is a pun, the context in which this joke is funny was never present. To understand the joke, one needs to know that working in the office often means working with Excel. This knowledge is not present to everyone and would only be understand in the current time, when Microsoft products are widely used. In further work the model can be used to compare the views of people from different cultural backgrounds on particular kinds of humour or metaphor. Our results show that further work is required to develop better representations of context, particularly for evaluating metaphors. Our analysis considered a relatively narrow type of humor – puns – which we will expand upon in future work. Another important direction is to develop suitable representations for longer texts where sentence embeddings may be less representative of creative language characteristics.

Acknowledments

We would like to thank Pia Snella for proof(read)ing the social implications of our model.

Parts of this research were conducted using the supercomputer MOGON 2 and/or advisory services offered by Johannes Gutenberg-University Mainz (hpc.uni-mainz.de), which is a member of the AHRP (Alliance for High Performance Computing in Rhineland-Palatinate, www.ahrp.info) and the Gauss Alliance e.V.

The authors gratefully acknowledge the computing time granted on the supercomputer MOGON 2 at Johannes Gutenberg-University Mainz (hpc.uni-mainz.de).

This project was partly funded by the project RMU (Rhine-Main Universities) Network for Deep Continuous-Discrete Machine Learning (DeCoDeML) of RMU-Initiativfonds and by the "Data Analytics for the Humanities" grant by the Hessian Ministry of Higher Education, Research, Science and the Arts.

Author Contributions

JS developed the DirectRanker for text ranking. MK adapted the Gaussian Process layer for the use with the DirectRanker. The experiment design was done by JS, MK, ES and KS. JS performed and analyzed the experiments shown in Table 1 and Figure 2 with the help of MK. The manuscript was written by JS, MK, ES and KS. IG and SK supervised the study and reviewed the manuscript.

References

[Cabezas-García and Reimerink 2022] Cabezas-García, M., and Reimerink, A. 2022. Cultural context and multimodal knowledge representation: Seeing the forest for the trees. In *Frontiers in Psychology*.

[Cao and others 2007] Cao, Z., et al. 2007. Learning to rank: From pairwise approach to listwise approach. In *ICML*.

[Chen and Ritter 2020] Chen, Y., and Ritter, A. 2020. Model selection for cross-lingual transfer using a learned scoring function. In *arXiv:2010.06127*.

[Cooper and others 1992] Cooper, W. S., et al. 1992. Probabilistic retrieval based on staged logistic regression. In *ACM SIGIR*.

[Devlin and others 2018] Devlin, J., et al. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. In *arXiv:1810.04805*.

[Do Dinh and others 2018] Do Dinh, E.-L., et al. 2018. Weeding out conventionalized metaphors: A corpus of novel metaphor annotations. In *EMNLP*.

[Han and others 2020] Han, S., et al. 2020. Learning-to-rank with BERT in TF-Ranking. In *arXiv:2004.08476*.

[Köppel and others 2019] Köppel, M., et al. 2019. Pairwise learning to rank by neural networks revisited: Reconstruction, theoretical analysis and practical performance. In *ECML*.

[Mao, Lin, and Guerin 2019] Mao, R.; Lin, C.; and Guerin, F. 2019. End-to-end sequential metaphor identification inspired by linguistic theories. In *ACL*.

[Mikolov and others 2013] Mikolov, T., et al. 2013. Distributed representations of words and phrases and their compositionality. In *NeurIPS*.

[Miller and others 2017] Miller, T., et al. 2017. Detection and interpretation of English puns. In *SemEval*.

[Nogueira and others 2019] Nogueira, R., et al. 2019. Multi-stage document ranking with BERT. In *arXiv:1910.14424*.

[Press 2011] Press, A. 2011. Research and methods. In *Encyclopedia of Creativity (Second Edition)*.

[Press 2017] Press, A. 2017. Understanding creativity in the performing arts. In *Creativity and the Performing Artist*.

[Qiao and others 2019] Qiao, Y., et al. 2019. Understanding the behaviors of BERT in ranking. In *arXiv:1904.07531*.

[Reimers and Gurevych 2019] Reimers, N., and Gurevych, I. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

[Simpson and others 2019] Simpson, E., et al. 2019. Predicting humorousness and metaphor novelty with Gaussian process preference learning. In *ACL*.

[Skelton 2021] Skelton, C. 2021. Comedy by numbers — a comedy-writer's thoughts on algorithmic approaches to humour. In *ICCC*.

[Steen and others 2010] Steen, G., et al. 2010. A method for linguistic metaphor identification. from MIP to MIPVU. In *CELCR*.

[Sullivan 2016] Sullivan, K. 2016. Integrating constructional semantics and conceptual metaphor. In *Constructions and Frames*.

BIPLEX: Creative Problem-Solving by Planning for Experimentation

Vasanth Sarathy[1*] and **Matthias Scheutz**[2]
[1]SIFT, LLC
[2]Tufts University
vsarathy@sift.net, matthias.scheutz@tufts.edu

Abstract

Creative problem-solving in humans often involves real-world experimentation and observation of outcomes that then leads to the discovery of solutions or possibly further experiments. Yet, most work on creative problem-solving in AI has focused on solely mental processes like variants of search and reasoning for finding solutions. In this position paper, we propose a novel algorithmic framework called BIPLEX that is closer to how humans solve problems creatively in that it involves hypothesis generation, experimentation, and outcome observation as part of the search for solutions. We introduce BIPLEX through various examples in a baking domain that demonstrate important features of the framework, including its representation of objects in terms of properties, as well its ability to interleave planning for experimentation and outcome evaluation when execution impasses are detected, which can lead to novel solution paths. We argue that these features are essentially required for solving problems that cannot be solved by search alone and thus most existing creative problem-solving approaches.

Introduction

Suppose you need to tighten a screw but do not have a screwdriver. Among the available objects you can use are a coin and pliers. After short reflection, you grip the coin with the pliers turning the "pliers-cum-coin" combination into a makeshift screwdriver. Psychologists call this "creative problem-solving" (Maier 1930).

While this solution might have been easy for you, it would have been much more difficult for a novice or a child, and practically impossible for current state-of-the-art AI systems. There several reasons for this difficulty. For one, an agent would need to determine the relevant properties of a screwdriver that makes it the appropriate tool to operate on screws: that it has a flat tip that fits into the screw's slot, and that the fit is tight enough so that if the screw driver were rotated, the screw would rotate with it, in addition to noting that a rotational downward movement is needed with a certain amount of force and that the screwdriver provide a handle that makes gripping it and applying the force easier. This analysis of the properties of the screwdriver can then guide the search for objects and their properties that could be used as a substitute. While one important *insight* is to notice that the coin has the property of fitting tightly into the slot, it does not have the property of providing enough of a grip to apply the necessary forces to generate the required rotational movement. The additional *insight* then is that the pliers which has a much better handle can be used to grip the coin tightly and thus establish a rigid connection so that when the pliers is turned and push downward, the coin is equally turned and pushed downward, turning the screw.

From the example it should be apparent why current systems cannot perform such feats: they would have to integrate detailed perception and affordance inferences with common sense knowledge, property-based analysis and planning, hypothetical reasoning and simulation, and potentially experimentation and on-the-fly learning (e.g., how to best grip the coin). While providing a system that can do all of the above is unrealistic at this point, we can nevertheless make progress by focusing on important constituents of such as system. This is what we set out to do for this position paper, namely take a fresh look a planning for crafting objects like make-shift tools based on object properties.

We will introduce a novel planner called BIPLEX which has several important properties for this type of creative problem-solving: (1) it represents objects in terms of their properties and affordances, and can thus (2) handle unknown objects to the extent that it can capture them in terms of known properties, and most importantly, it can plan to craft novel objects based on property requirements. We demonstrate how BIPLEX can handle goals involving the crafting of novel objects based on property specifications. While for most planners a direct comparison to BIPLEX is not possible (because they cannot handle properties or unknown objects), we sketch out how even for goals that regular planners like Fast-Forward (FF) (Hoffmann 2001) can handle, BIPLEX can lead to significant speedups. Finally, we conclude by discussing some limitations of BIPLEX as well as future research questions in creative problem-solving.

Background and Related Work

Problem solving in humans typically involves deliberate and conscious processing that advances a solution step by step. Insight, on the other hand, is believed to involve a "sudden" and unexpected emergence of an obvious solution or strategy sometimes accompanied by an affective "Aha!" experience which is why solvers often find it difficult to consciously explain how they generated a solution in a sequential manner. MacGregor et al. proposed the

Criterion for Satisfactory Progress Theory (CSPT), which is based on Newell and Simon's original notion of problem solving as being a heuristic search through the problem space (MacGregor, Ormerod, and Chronicle 2001). The key aspect of CSPT is that the solver is continually monitoring their progress with some set of criteria. Impasses arise when there is a criterion failure, at which point the solver tries non-maximal but promising states. Ohlsson et al.'s *Representational Change Theory* (RCT), on the other hand, suggests that impasses occur when the goal state is not reachable from an initial problem representation (which may have been generated through unconscious spreading activation) (Ohlsson 1992). To overcome an impasse, the solver needs to restructure the problem representation through (1) elaboration (noticing new features of a problem), (2) reencoding (fixing mistaken or incomplete representations of the problem), and (3) changing constraints (believed to involve two sub-processes of constraint relaxation and chunk-decomposition). Ollinger's extended RCT is a dynamic and iterative or recursive process that involves repeated instances of search, impasse and representational change (Oellinger, Jones, and Knoblich 2014; Oellinger et al. 2017): a solver first forms a problem representation and begins searching for solutions; when an impasse is encountered because no solution can be found, the solver must restructure or change the problem representation and once again search for a solution, thus combining heuristic searches, hill climbing and progress monitoring with creative mechanisms of constraint relaxation and chunk decomposition to enable restructuring.

Another related theory of creative problem solving views insight as the retrieval of an analogy from long-term memory using spreading activation (Langley and Jones 1988). This view depends on having sufficient prior experience and thus knowledge encoded such that an analogical mapping can be established.

Different from the above proposals, we are making the core claim that creative problem solving is not just a mental exercise, one that can be approached with "searching" some problem space alone, but that is essentially involves experimentation when impasses due to knowledge limitations are reached (e.g., when no plan for goal accomplishment can be found or when the execution of a plan fails with no explanation for what went wrong). Only through formulating hypotheses that lead to new experiments and observation of the outcome of these experiments is it possible for an agent to augment its knowledge base with novel facts about objects, their properties, and their affordances, which, in turn can be used by the planner to find different routes to the goal states.

The view that creative problem solving requires the extension of one's concepts has be echoed recently by (Gizzi et al. 2020) who define creative problem solving 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." Yet, their proposal only involves combinatorial methods (combining existing concepts), transformational methods (changing conceptual representations), and exploratory methods (searching the concept space). As we will show below, these methods are insufficient for discovering solutions to even simple problems without experimentation.

(Freedman et al. 2020) use analogical reasoning to find a substitution for a missing resource that is needed in a plan that accomplishes the goal. However, all knowledge about potential resource substitution candidates needs to be provided to the planner in order to accomplish the reasoning, there is no discovery of potential resource candidates involved. Analogical reasoning is solely used to find candidates that are close in property structure. In contrast, our approach hypothesizes potential substitutions based on common properties (which could be based on analogical reasoning as well) and devises experiments to test whether they, in fact, have the desired properties. Consequently, it does not require advance knowledge about all possible substitutions but it can discover them.

The approach closest to our approach of planning experiments to learn more about the domain is (Lamanna et al. 2021) which attempts to learn the actions and thus operators in a planning domain under certain assumptions about the structure of pre- and post-conditions in a deterministic setting. The algorithm starts with a set of given operators but without any knowledge about their pre- and post-conditions and systematically performs experiments to determine the exact conditions. This is done by initially starting with a set of pre-conditions consisting of all atoms that can be formed using a set of variables and predicates and an empty set of post-conditions for all operators, observing the outcome of performing an applicable operator in the current state and updating the pre- and post-conditions based on what is true in the predecessor and the successor states. However, their learning algorithm does not deal with transformative actions that change the objects in the planning domain, nor does it deal with property representations of objects in pre- or post-conditions, and hence cannot learn object substitutions in actions as it requires constants denoting all objects ahead of time. Moreover, since the goal is to learn the complete domain model, it does so without any particular preference for a given operator or particular state; all that matters is that the planner pick a state to explore that could potentially refine the pre- and post-conditions and as long as such states exist that the planner can get to using a finite state machine model of the domain it has learned so far that has a state size exponential in the number of facts, i.e., grounded predicates – clearly that latter makes this approach intractable for domains with a large number of objects. In contrast, our approach explores only those actions that could advance its goal to produce objects that are not available and thus does is not impacted by large numbers of objects in the domain.

Another related experimentation system implemented on a robot focuses on finding matching objects that can be used for creating tools based on a set of given reference tools (Nair et al. 2019). While this approach does not discover the tool per se or infers properties of tools needed, it demonstrates how an embodied agent could use its perceptual apparatus to determine properties of objects that are similar enough to desired properties, use those objects to assemble tools and evaluate those tools on actions that use them.

As such, our planner would be synergistic with the robot system in that it would provide the robot with tool properties that the robot can then assemble and evaluate, i.e., the robot would function as the experimenter carrying out the actions proposed by our planner in order to determine whether the resulting objects have the desired properties. More recently, there have been advances, in the robotics domain, for finding object substitutions based on properties (Fitzgerald, Goel, and Thomaz 2021). This work proposes the use of constraints (tool shape, segments, and visual features) that will be useful in exploring and evaluating other potential tool candidates. The approach suggests the discovery of new tools, however, what is missing then is building on this knowledge to compose and construct new tools using available resources.

The general idea of learning domain knowledge through experimentation is not new, and certainly not being claimed as such in this paper. Early work in the symbolic AI literature explored how agents can adjust and improve their symbolic models through experimentation. Gill proposed a method for learning by experimentation in which the agent can improve its domain knowledge by finding missing operators (Gil 1994). The agent is able design experiments at the symbolic level based on observing the symbolic fluent states and comparing against an operator's preconditions and effects. Other approaches (to name a few: (Shen 1989; Shen and Simon 1989; Mitchell, Keller, and Kedar-Cabelli 1986; Yang, Wu, and Jiang 2007; Aineto, Jiménez, and Onaindia 2019; Cresswell, McCluskey, and West 2013; Hogg, Kuter, and Munoz-Avila 2010; Sussman 1973)), comprising a significant body of literature, have explored learning from experimentation, recovering from planning failures, refining domain models, and open-world planning. Recently, (Musliner et al. 2021) proposed a planning system capable of hypothesis generation, model-modification and evaluation using a library of domain-independent heuristics useful to help agents accommodate novelties in the environment. What is missing from these approaches is a solution for handling transformative actions (like object construction or crafting) where new object types are generated during execution, which are then needed to solve the problem at hand. As we saw in the coin-plier example presented earlier, and as we will demonstrate in the rest of the paper, selecting object substitutions, composing them together to generate entirely new types of objects and reasoning about these compositions to re-plan and revise execution is needed for creative problem solving.

Introducing BIPLEX

In this section we introduce our novel BIPLEX (**bi**nd types, **pl**an and **ex**ecute) approach to creative problem-solving.[1] Rather than starting with abstract principles, we will motivate its core ideas and ways to solve problems by working through examples in a baking domain. We selected the baking domain because it is a familiar domain with a combination of navigation, manipulation and transformative actions involving mixing items to produce new products, thus allowing ample room for creativity through creating novel products and resourceful substitution of unavailable items. Tracing through the operation of BIPLEX then will show the synergistic effects of (1) open-world property-based planning and (2) experimentation through plan execution with subsequent (3) rule learning, plan refinement and replanning based on the outcome of hypothesized experiments. In addition to highlighting the involved principles, we will also point the technically interested reader to place in the pseudocode that accomplish the various steps in the process.

We start with a departure from the classical planning framework in how we represent objects by allowing object definitions in terms of functional and material properties (the reason for this will become clear in a bit). In the baking domain this means that we will describe the various baking ingredients such as eggs, egg whites, sugar, flour, etc. as well as utensils such as bowls, spoons, frying pans, etc. in terms of their defining properties (but critically without attempting to give sufficient conditions, only necessary ones for the baking domain). For example, we might describe "egg yolks" as "yellow and liquid and containing fatty acids" and "egg whites" as "transparent and liquid and containing protein" (leaving out the color as they are transparent).[2] Similarly, we might describe actions in terms of pre-conditions ensuring applicability, action signature (i.e., the action together with its arguments), and the expected post-conditions when execution succeeds.[3]

Now consider a goal for the agent to make *egg batter*, an important step in pancake-making. Egg batter, in our environment is a class of objects of type **`yftl`**.[4] Unlike most classical planners, BIPLEX allows for specifying lifted-goals[5] of the form (`have ?x-`**`yftl`**), with variable and type declaration. Such a goal can be satisfied if there exists a literal in the current state that evaluates to true, where the literal's name unifies to the name of the goal and the literals argument is a constant that has the same type or subtype[6] as that specified in the goal. Most classical planners require a grounded goal, which means, the agent would need to instantiate the constant associated with a desired type, which, in turn, means the agent would need to, apriori, instantiate all the types it might be able to craft, in case, it will later need to plan to craft one of those. This becomes intractable very quickly, once one realizes that in many real-world settings, there are many instances of each object type – many tomatoes, many teaspoons of sugar, etc. We will explore various cases, each of increasing difficulty to describe some of the capabilities of BIPLEX .

[1] BIPLEX has been fully implemented and the code can be found at: https://github.com/vasanthsarathy/biplex

[2] We can represent each property with a single character, e.g., "y" for yellow and "t" for protein and "f" for fatty acids

[3] This is generally how domains are represented in classical AI planning.

[4] A "type" in our formulation is merely a sequence of characters, each specifying a property.

[5] "lifted" here means goals that do not have grounded constants in their terms, but instead have variables

[6] Subtypes might be represented as a type hierarchy provided to planners to facilitate variable binding

Problem-Solving with Transformative Actions

Consider the goal, (have *?x*-**yftl**) corresponding to egg batter. Let us assume that the BIPLEX agent has all the resources it needs to make egg batter **yftl**. That is, it has the domain specification[7] that contains several navigation and manipulation actions along with several transformative actions like the following (variables start with a "?" and italicized, types are in bold, action names are in bold and italic):

```
(:action mixeggbatter1
:parameters (?y - yftl ?x1 - yfl ?x2 - tl ?x3 - wl
    ?z - rmc)
:precondition (and (have ?x1) (have ?x2) (have ?x3)
    (have ?z))
:effect (and  (have ?y)  (not (have ?x1))
    (not (have ?x2)) (not (have ?x3)) ))
```

This action ***mixeggbatter1*** requires the agent to have ingredients of type **yfl** (egg yolk), **tl** (egg white), **wl** (milk), and an object of **rmc** (bowl) to mix the ingredients. At the completion of then action, the ingredients are consumed (except the bowl) and the agent is left with an egg batter object **yftl**. This action is a "transformative action" in the sense that a new object type is created and some ingredients are consumed and cease to exist as objects in the problem space. The agent may need to sequence a plan to find and collect these ingredients and therefore must interleave navigation and manipulation with mixing or crafting. To achieve this goal with a state-of-the-art classical planner (like FF (Hoffmann 2001)), we would need to instantiate all constructable objects, which in this case includes generating a constant symbol for the egg batter. An FF agent would need to also ground the goal to this constant object, and then proceed to generating a complete plan before it can begin any execution. The FF agent, thus begins with an domain specification and a problem specification[8] with all possible constants of all possible types and a grounded goal. In most classical offline planning scenarios, execution does not begin until the FF agent has ground all the variables from all the types, completed planning and produced a plan.

Like the FF agent, the BIPLEX agent is also given a domain (action schema) specification containing navigation, manipulation and transformative actions. However, unlike the FF agent, the BIPLEX agent is not given a complete problem specification. Instead, we embed the agent in an environment allowing it to plan and execute in an interleaved manner. The agent scans the environment and observes objects present in the current initial state along with their respective types. Note, the agent cannot observe objects with types that have not yet been generated by transformative actions. So, initially, the BIPLEX agent does not observe eggbatter1 as it does not yet exist. In addition to a domain specification, the BIPLEX agent is also provided a lifted goal (have *?x*-**yftl**).

From the domain specification, the BIPLEX agent generates (1) a stripped-down domain specification, and (2) a tree with nodes representing inputs and outputs of the transformative action and nodes representing the name of the transformative actions. The stripped-down action schema contains all the transformative action schemas, but stripped-down to only contain their output types, any preconditions, and any types that the agent believes will not be transformed by the action. For example, the ***mixeggbatter1*** will be stripped-down to ***hyp-mixeggbatter1***[9] :

```
(:action hyp-mixeggbatter1
:parameters (?y - yftl ?z - rmc)
:precondition (and (have ?z))
:effect (and  (have ?y) )
```

The BIPLEX agent first grounds the lifted-goal with a gensym, a hypothetical constant symbol (e.g., (have hyp-yftl-a9b88541)), and then generates a problem specification based on information it can observe from the initial state. It then attempts to generate a "plan sketch" using the stripped-down domain specification and problem specification (line 12, Algorithm 1). The agent can do this at very low computational cost as the stripped transformative actions have far fewer parameters and no preconditions. Moreover, using the stripped-down domain allows BIPLEX to reason about transformative actions that have ingredients that themselves are products of other transformative actions. Without any preconditions, BIPLEX can essentially generate a list of resources and intermediate products it will need, a shopping list of sorts, if you will. If the plan sketch does not contain any stripped-down actions or any hypothetical constant symbol names, the BIPLEX agent simply executes the plan (lines 14-16, Algorithm 1). This occurs when the actions are primarily navigational or involve manipulation. If, however, the plan contains hypothetical symbols or transformative actions, it will turn to the crafting tree to construct the types it needs (lines 25-35, Algorithm 1). Instead of having to *apriori* generate all the constants for all possible types as is required for FF, BIPLEX only generates symbols for types as and when it needs them. At the crafting tree, it finds the type that it needs to construct, then traverses the crafting tree to find the associated transformative action and its linked input ingredients.

BIPLEX uses two operations – **PROVE** (Algorithm 2) and **GROUND** (Algorithm 3) – to bind types to ground constants as it traverses the crafting tree. In our running example, the BIPLEX agent will find a plan-sketch (Algorithm 1): (***pickup*** bowl1)[10] and then (***hyp-mixeggbatter1*** hyp-yftl-a9b88541 bowl1).[11] This two-step plan-sketch cannot be executed as there are hypothetical constants as well as stripped-down actions. The agent will attempt to "prove" the existence of an object of type **yftl** by "grounding" any action that is a predecessor to type **yftl** in the crafting tree, namely any action that constructs **yftl** (line 2, Algorithm2). In this example, we

[7] This is a PDDL 1.2 representation of an action schema usually contained in a domain.pddl file that is provided to a planner.

[8] Also provided to the planner as a problem.pddl file

[9] We adopt the convention of representing stripped-down action names with the prefix "hyp-"

[10] We use LISP notation as is customary in AI planning for representing actions with action name followed by parameters within a list

[11] As part of Stanford's MOLGEN project, Mark Stefik echoed the idea of generating relevant "skeletal plans and then refining them" (Stefik 1981).

only have one action ***mixeggbatter1***, and so the agent will attempt to "ground" it. Grounding an action in the crafting tree involves (recursively) proving all of its predecessor types (lines 2-10, Algorithm 3). Here, the action has several predecessor types (which are the input ingredients to the mix action) including **`yfl`** (egg yolk), **`wl`** (milk) and **`tl`** (egg white). For each of these resources, BIPLEX will interleave planning and execution to first acquire milk by planning and executing (***gofromto*** `bowl1 milk2`), (***pickup*** `milk2`), and then planning and executing (***gofromto*** `milk2 yolk2`), (***pickup*** `yolk2`), and then (***gofromto*** `yolk2 eggwhite2`), (***pickup*** `eggwhite2`). As it acquires each ingredient it proves the corresponding type node in the crafting tree. Upon proving all the predecessor nodes of the action, the BIPLEX agent then is ready to perform the transformative mix action itself (lines 11-26, Algorithm 3). To do so, it generates a new domain specification containing the navigation and manipulation actions along with a single, fully specified transformative action, which in this case is ***mixeggbatter1***. The agent generates a new planning problem specification with the goal of (`have` *`?x`*`-`**`yftl`**). Remember, the agent still needs a bowl, which it will acquire next. Upon completion of this plan, a new constant appears in the agent's observations (`eggbatter1`) and several other constants disappear. We perform some bookkeeping to replace the hypothetical symbol for eggbatter with the one from the environment, as well as removing objects that were consumed. We maintain a strict separation of the agent and the environment, so the agent does not know about an object unless it is perceivable in the environment.

The approach of breaking down the problem is more intuitive, allowing for not only easier debugging, but also significantly faster performance. This is the case, because each planning instance during grounding only contains a single transformative action along with other navigational and manipulation actions. Moreover, only those types are instantiated that are needed per the crafting tree.

Next we will explore the benefit of representing classes of objects or types as a conjunction of properties. We have alluded to this earlier that creative problem-solving involves reasoning about object properties themselves. We will next discuss how this reasoning process can be computationalized.

Creative Problem-Solving with Hypothesis Generation and Testing

Thus far, we have shown that BIPLEX can solve problems that classical planners can also solve, but BIPLEX solves them faster and is able to handle transformative actions as well as lifted specifications more naturally. We now turn to discussing how BIPLEX is also creative. Continuing with our running example, consider what happens if there is no egg yolk. Classical planning systems like FF would fail as no plan exists given the domain and problem – without yolk, the agent cannot make egg batter. Using findings from human problem-solving we propose that the agent should consider other objects with similar, possibly overlapping properties. We operationalize this creative mechanism as follows.

In addition to **PROVE** and **GROUND**, BIPLEX agents have the ability to **PLAY** that is, "hypothesize" and "test" resource substitution ideas. At the core of this capability is the ability to track object properties. The key idea here is that the BIPLEX agent is able to compose available types to generate ideas for new novel types. The agent does not know, apriori, if these combinations will work or even if relevant properties will persist after combination. The best an agent can do is assume that certain properties exist and experiment and make discoveries. Based on the result of the experiment, the agent is able to gain additional knowledge with which to derive improved future hypotheses. We will now walk through an example of how the BIPLEX agent accomplishes this task.

First, consider simpler goal of having a egg yolk (`have` *`?x`*`-`**`yfl`**) when there isn't any available in the environment. Moreover, there is no transformative action available to make egg yolk. As noted earlier, BIPLEX will first try to generate a plan-sketch using the stripped-down domain specification (Algorithm 1). However, the planner will fail, as we might expect. Upon failure, BIPLEX will enter "creative mode" and first attempt to hypothesize different ways it can compose together objects (lines 18-24, Algorithm 1). BIPLEX does this by generating a set of available interesting types. This is a set of types, where each type (which is represented as a conjunction of properties) intersects with the target type **`yftl`** in terms of properties. Thus, if the environment contained the following objects: milk **`wl`**, water **`l`**, yogurt **`wftl`**, applesauce **`yf`** then these would all be intersecting types as they contain one or more properties that are also properties in our target type **`yftl`**. BIPLEX then generates a power set of these intersecting types (line 1, Algorithm 4). Each element of the power set represents a possible composition of multiple types that, in theory, could be combined to generate the target type. BIPLEX also filters this power set to only include those elements that when combined possess all the properties of the target type. Thus, yogurt and applesauce together have at least the properties y-f-l, and so are included. However, yogurt and water together do not have the property "y", so this composition is not included in the power set. This filtered power set is a complete set of hypotheses available to the BIPLEX agent to experiment with.

For each hypothesis in the set of candidate hypotheses, BIPLEX generates a novel (generic) action (let's call it ***mix1***) that is intended to serve as a generic template with which to allow the agent to try to mix objects. not sure how to do this generally. The agent uses this template to generate a domain and problem specifications to allow it to plan and execute the particular experiment (lines 10-12, Algorithm 4). Upon completion, BIPLEX reviews the new state and compares it against the previous state. It declares the experiment a success if the new state contains an object of a novel type that did not exist in the prior state. To compare states, the agent inspects the type signature (conjunction of properties) of this novel type to ensure that it possesses all the desired properties of its target type **`yfl`**.[12] If so, the hypothesis-testing

[12] It is worth noting that we are assuming that the agent can observe all the types be it directly or indirectly via inference or measurement instruments.

phase is declared complete, and the agent returns to generating plan-sketches followed by planning, crafting and execution as described earlier. When the agent discovers the new type, it does not yet know if this will be all it needs for the rest of the plan. If it encounters another impasse (say a missing resource), it will revisit hypothesis-testing then to troubleshoot and make new discoveries. Now, if after performing the experiment, the agent does not observe a state change (i.e., no new types were constructed), the agent will try the next hypothesis. If none of the hypotheses work, the agent will give up and end problem solving. It is conceivable, although we have not implemented this capability, this may be when and where the agent will need to discover or consider a new property of the environment. Human creative problem-solving involves this capability, and often solutions to insight problems are found when a property is found to be relevant, but previously thought to be irrelevant (Sarathy 2018; Sarathy and Scheutz 2018). The classic example of this is the three-bulb insight problem in which the solver must discover an unexpected property of the light bulb to solve it.

Thus, in our example of having egg yolk, the agent found 79 hypotheses, tested two of them before it found one that works. First it considered combining yogurt **`wftl`** and oil **`ly`**. While these two when combined possess all the properties of egg yolk **`yfl`**, when tested, there is no observed state change. The reason for this is that in our testing environment, we did not encode any new type for combining these two elements. The agent was not told this apriori, but instead discovered it during hypothesis testing. The next hypothesis the agent tried was to combine yogurt **`wftl`** with applesauce **`yf`**. When tested, the BIPLEX agent discovered that it produces a new type, **`ylft`**, which is really a wet batter without any eggs. The agent declares success here because **`ylft`** contains the properties “y”, “f” and “l”, which was what defined egg yolk, the target type. Now, clearly this is not the same as egg yolk and it might seem a bit strange to call this a success, however, it is worth noting that in this example, we did not equip the environment with any dynamics for making egg-yolk. So, the agent found the best answer with the resources it had on hand. What is particularly interesting about this new type **`ylft`** is that it contains the same properties (in a different order) as egg batter. The agent might have discovered this as a substitute for egg batter if it were planning to make pancakes and not just egg yolks, is that this new type allows it to skip the step of making egg batter and move to the other steps of pancake making. This would simplify the agent’s planning and execution as it would not need to seek out and acquire all the other ingredients for making egg batter as it already has what it needs, a discovery it made while making something else, namely egg-yolk. We next discuss how the agent might creatively make egg batter with no egg-yolk if it did not make this discovery.

Consider the goal of having egg batter (`have ?x-`**`yftl`**), as we discussed previously. Unlike when we trying to make egg yolk, here, the initial attempt at generating plan-sketch will not fail. The agent will generate a plan-sketch: (***pickup*** `bowl1`), (***hyp-mixeggbatter1*** `hyp-yftl-ac7adcdf`). Realizing it needs to construct an object of type **`yftl`**, it will search the crafting tree. The crafting tree indeed has a node for egg batter, which it will traverse and identify the ingredients needed, the first of which is milk **`wl`**. The agent will then plan and execute to go and get milk. Once it does that the next ingredient for the agent is to acquire egg yolk **`yfl`**. As we mentioned above, since there is no egg yolk, the agent will enter creative mode, generate hypotheses (79), try and experiment until it finds one that works. Let’s say the one it finds is combining water **`l`** and applesauce **`yf`** to thereby generate diluted applesauce **`lyf`**. At this point, we know that diluted applesauce is not the same as egg yolk and is only potentially a viable substitute for it in egg batter. But, the agent marches on and continues the process of making egg batter and plans and acquires egg whites **`tl`**. Once it has “proved” all the ingredients or predecessors to the action node ***mixeggbatter1***, it is ready to “ground” the action, and “prove” the type **`yftl`** for egg batter. However, this plan will fail because it cannot make egg batter with diluted apple sauce. At this point, the BIPLEX agent once again enters creative mode to find a substitute for egg batter. Amongst the many hypotheses, one that works is one that requires the use of the newly created diluted apple sauce to make non-egg egg batter **`ylft`**. The agent then declares success in making a substitute and ends problem solving. Again, this may not be a viable substitute for downstream use, but in this particular instance of making no-egg-yolk-pancakes, this substitute works just fine.

Thus far we have discussed creative resource substitution, when an agent is attempting to acquire a known type, say egg batter or egg yolk. These are types listed in the agent’s own domain, and types for which the agent has domain knowledge about the types of actions that can performed with them and the fluents that apply to them. However, the approach underlying BIPLEX can be used to pursue previously unknown goals. For example, if the agent is tasked with making diluted applesauce **`lyf`** or even making no-egg-yolk egg batter **`ylft`** or no-egg-yolk egg pancake **`vaftuy`**. The planning, execution, constructing, hypothesis generation and testing proceeds just as described before.

It should now be clear why we need to use properties to represent object types instead of using those types themselves, or at the very least properties in additions to types; for without explicit property representation we would not be able to formulate the kinds of hypotheses we need in order to handle novel situation and learn substitutions through experimentation. It will also not be able to find plans that ultimately require the creation of novel objects or substances.

Discussion and Limitations

As we discussed earlier, most existing work, particularly in AI, approaches creativity and creative problem-solving as a mental search problem for possibilities and associations. This view, however, neglects the critical role of real-world experimentation and its potential for gaining insight from observations during the experimentation process. As (Sarathy 2018) has noted, there is evidence in human neuroscientific studies that the environment, and the human’s interaction with the environment is crucial to the creative

```
Algorithm 1: SKETCH()
Input: $goals$ {global variable}
Input: $completed$ {global variable}
while $\exists\ goals$ do
    $goal \leftarrow goals$.pop()
    if $goal \in s_0$ then
        $completed$.append($goal$)
        return True, $s_0$
    end if
    if $goal \in completed$ then
        return True, $s_0$
    end if
    $objects \leftarrow$ observe-objects()
    $\mathcal{P} \leftarrow$ construct-problem($goal, s_0, objects$)
    $\pi \leftarrow$ plan($\Sigma^*, \mathcal{P}$)
    $\mathcal{O}^* \leftarrow$ is-executable($\pi$)
    if $\pi \neq \emptyset$ and $\mathcal{O}^* = \emptyset$ then
        completed.append($goal$)
        return execute($\pi$)
    end if
    if $\pi = \emptyset$ then
        $status, s_1 \leftarrow$ play($goal$)
        if $status =$ False then
            return False, $s_0$
        end if
        return True, $s_1$
    end if
    if $\mathcal{O}^* \neq \emptyset$ then
        $objects \leftarrow$ get-objects-to-construct($\mathcal{O}^*$)
        for $object \in objects$ do
            $type \leftarrow$ get-type($object$)
            $status, s_1 \leftarrow$ PROVE($type$)
            if $status =$ False then
                return False, $s_0$
            end if
        end for
        return True, $s_1$
    end if
    return False, $s_0$
end while
return True
```

process. For new knowledge needs to enter the mind somewhere and we claim that this interaction with environment must be interleaved with mental search and reasoning processes. Moreover, there is a mutually-supportive aspect of this relationship between environmental experimentation and mental search, whereby when the agent notices something interesting in the environment, it may trigger new mental search and reasoning processes that can lead to new experiments to further understand the observation, and so on.

The BIPLEX approach to planning is thus different from most existing planning approaches that rely solely on grounding action schemas to every object provided in a problem instance and finding a path in the search space to the goal. Although BIPLEX calls a planner as part of its operation, the planning is limited to small instances enabling BIPLEX to handle large problem instances with a large number of objects. Creativity in the real-world involves the ability to reason about large number of possible objects, ill-defined problem statements and partial knowledge. These real-world conditions prevent standard planners from being easily applied in creative settings. Here, we suggest that BIPLEX can serve as a "wrapper" over fast state-of-the-art planners, one that can handle these real-world constraints more effectively. Moreover, it is unclear how BIPLEX should decide which properties are relevant to a particular problem instance. BIPLEX was designed with "fluidity" in mind, which can be seen in how it dynamically defines and redefines new problem instances as it approaches its goal. This

```
Algorithm 2: PROVE($type$)
Input: $type$ {An object type}
Input: $tree$ {global variable}
$s_0 \leftarrow$ observe()
$actions \leftarrow$ get-predecessors($tree, type$)
if $actions = \emptyset$ then
    return False, $s_0$
end if
while $actions \neq \emptyset$ do
    $action \leftarrow actions$.pop()
    $status, s_1 \leftarrow$ GROUND($action$)
    if $status =$ True then
        return True, $s_1$
    end if
end while
return False, $s_0$
```

```
Algorithm 3: GROUND($action$)
Input: $action$ {An action name}
Input: $tree$ {global variable}
Input: $grounded$ {global variable}
$types \leftarrow$ get-predecessors($tree, action$)
for $type$ in $types$ do
    $s_0 \leftarrow$ observe()
    $goal \leftarrow$ construct-goal-literal($type$)
    $goals$.push($goal$)
    $status, s_1 \leftarrow$ SKETCH()
    if $status =$ False then
        return False, $s_0$
    end if
end for
if $action \notin grounded$ then
    $grounded$.append($action$)
    $\Sigma \leftarrow$ create-domain($\Sigma^-, action, tree$)
    $type \leftarrow$ get-successor$action, tree$
    $goal \leftarrow$ construct-goal-literal($type$)
    $status, s_1 \leftarrow$ plan-and-execute($goal, \Sigma$)
    if $status =$ True then
        $goals$.push($goal$)
        return SKETCH()
    end if
    $status, s_1 \leftarrow$ PLAY($goal$)
    if $status =$ True then
        return True, $s_1$
    end if
    return False, $s_0$
end if
return True, $s_1$
```

Algorithm 4: PLAY($goal$)

```
Input: $goal$
$combinations \leftarrow$ get-intersecting-properties($goal$)
$hypotheses \leftarrow \emptyset$
for $comb \in combinations$ do
    if $goal$.type $\subseteq comb$ then
        $tree \leftarrow$ make-tree($comb$, $goal$)
        $hypotheses$.append($tree$)
    end if
end for
$s_0 \leftarrow$ observe()
for $hypo$ in $hypotheses$ do
    $\Sigma \leftarrow$ create-domain($\Sigma^-$,generic,$hypo$)
    $status, s_1 \leftarrow$ plan-and-execute($goal$, $\Sigma$
    if $status =$ True then
        if $s_0 \overset{\text{type}}{=} s_1$ then
            continue
        else
            $completed$.append($goal$)
            return True, $s_1$
        end if
    end if
end for
return False, $s_1$
```

fluidity, however, has many dimensions beyond what we have already shown: from selection of properties, granularity of representations, and selection of relevant objects to consider. In future work, we intend to study other dimensions of fluidity as we believe it can help lower complexity by reducing the particular problem instances for an underlying planner.

A limitation of the current approach is that the agent tries each hypothesis independent of any prior attempts. It is reasonable to expect that the agent should "learn" from each experiment and only try hypotheses that are likely to produce new information. (Lamanna et al. 2021) discuss the value of this idea in their online planner that learns a planning domain during execution. One avenue for future work is to consider how the agent could learn from an experience and use this knowledge in a different problem. We are not advocating that creative problem-solving be fully-unguided exploration. Instead, BIPLEX relies on a predefined set of relevant properties over which the object types are defined, and the agent itself is goal-directed, in that we have provided a planning goal that they agent must reach. We intend to explore, in future work, different strategies for guided experimentation, hypothesis generation, and representations for learned and acquired knowledge.

Another limitation is that BIPLEX executes each viable hypothesis until it finds one that works. However, there may be other constraints (such as cost, time, availability of materials, normative and ethical ones) that limit what is doable and acceptable to experiment with for the agent. In some cases it may thus be prudent to put a human in the loop and allow BIPLEX to suggest new ideas and give the human final decision-making authority as to whether to test a hypothesis or not, especially if it is likely that the human or even another agent might already know what will happen. In collaborative settings, the agent may essentially be able to eliminate hypotheses without even trying them.

BIPLEX generates hypotheses when it must construct novel types or when it needs to find a substitute to a known type and it declares success in its hypothesis testing when it finds a type that has all the desired properties of its target type. At this point, BIPLEX does not know if the newly found type will work for the rest of its plan. It is thus possible that somewhere downstream an action fails as a result of not having found the right substitute here. In such a case, BIPLEX should backtrack and revisit other successful hypotheses, which is currently not implemented.

Finally, BIPLEX assumes, during hypothesis generation, that a target type must be formed with a combination of other types that have intersecting properties. However, it is possible that non-intersecting types produce novel types. Take for example, combining salt and ice: salt is of type solid-granular-white and ice is of type solid-cold. When combined, salt melts the ice to produce water which is not a solid, does not contain granules, is not white and presumably is less cold than ice. So, if the target object is water of type clear-liquid, BIPLEX would not combine these two. Clearly, the possibility of combining known types to yield potentially novel types is an important direction for future work as it will allow the agent to expand its conceptual basis (and, of course, is it yet another interesting question of how the agent would be able to recognize the new type).

Conclusion

In this position we introduced the BIPLEX framework, a novel approach for creative problem-solving that uses property-based representations of objects to enable planning with transformative actions and object substitution in a tightly integrated hypothesis generation, experimentation, observation, and adaptation loop. By planning experiments and performing them to utilizing observations of their outcomes in the planning process, BIPLEX offers an approach for problem solving that goes beyond mere search-based methods and more closely mimics the process of scientific discovery which essentially involves experiments in the real world to try out theoretical predictions and confirm or reject hypotheses. In a next steps, we plan to evaluate the performance of BIPLEX and compare it to state-of-the-art planners to demonstrate that it can better handle large numbers of objects due to its property-based representations and that it can solve problems that other planners cannot solve due its ability to perform experiments.

Author Contributions

Vasanth Sarathy and Matthias Scheutz jointly developed and wrote the ideas and algorithms in this paper.

Acknowledgements

This work was funded by DARPA grant W911NF-20-2-0006.

References

Aineto, D.; Jiménez, S.; and Onaindia, E. 2019. Learning strips action models with classical planning. *arXiv preprint arXiv:1903.01153*.

Cresswell, S. N.; McCluskey, T. L.; and West, M. M. 2013. Acquiring planning domain models using locm. *Knowledge Engineering Review* 28(2):195–213.

Fitzgerald, T.; Goel, A.; and Thomaz, A. 2021. Modeling and learning constraints for creative tool use. *Frontiers in Robotics and AI* 8.

Freedman, R.; Friedman, S.; Musliner, D.; and Pelican, M. 2020. Creative problem solving through automated planning and analogy. In *AAAI Workshop on Generalization in Planning*.

Gil, Y. 1994. Learning by experimentation: Incremental refinement of incomplete planning domains. In *Machine Learning Proceedings 1994*. Elsevier. 87–95.

Gizzi, E.; Nair, L.; Sinapov, J.; and Chernova, S. 2020. From computational creativity to creative problem solving agents. In *Proceedings of the 11th International Conference on Computational Creativity (ICCC'20)*.

Hoffmann, J. 2001. Ff: The fast-forward planning system. *AI magazine* 22(3):57–57.

Hogg, C.; Kuter, U.; and Munoz-Avila, H. 2010. Learning methods to generate good plans: Integrating htn learning and reinforcement learning. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*. Citeseer.

Lamanna, L.; Saetti, A.; Serafini, L.; Gerevini, A.; and Traverso, P. 2021. Online learning of action models for pddl planning. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*.

Langley, P., and Jones, R. 1988. A computational model of scientific insight. In Sternberg, R. J., ed., *The Nature of Creativity: Contemporary Psychological Perspectives*. New York, NY: Cambridge University Press. 177–201.

MacGregor, J. N.; Ormerod, T. C.; and Chronicle, E. P. 2001. Information processing and insight: a process model of performance on the nine-dot and related problems. *Journal of Experimental Psychology: Learning, Memory, and Cognition*.

Maier, N. R. 1930. Reasoning in humans. i. on direction. *Journal of comparative Psychology* 10(2):115.

Mitchell, T. M.; Keller, R. M.; and Kedar-Cabelli, S. T. 1986. Explanation-based generalization: A unifying view. *Machine learning* 1(1):47–80.

Musliner, D. J.; Pelican, M. J.; McLure, M.; Johnston, S.; Freedman, R. G.; and Knutson, C. 2021. Openmind: Planning and adapting in domains with novelty. In *Proceedings of the Ninth Annual Conference on Advances in Cognitive Systems*.

Nair, L.; Shrivastav, N.; Erickson, Z.; and Chernova, S. 2019. Autonomous tool construction using part shape and attachment prediction. In *Proceedings of Robotics: Science and Systems*.

Oellinger, M.; Fedor, A.; Brodt, S.; and Szathmary, E. 2017. Insight into the ten-penny problem: guiding search by constraints and maximization. *Psychological Research* 81(5):925–938.

Oellinger, M.; Jones, G.; and Knoblich, G. 2014. The dynamics of search, impasse, and representational change provide a coherent explanation of difficulty in the nine-dot problem. *Psychological research* 78(2):266–275.

Ohlsson, S. 1992. Information-processing explanations of insight and related phenomena. *Advances in the Psychology of Thinking* 1–44.

Sarathy, V., and Scheutz, M. 2018. Macgyver problems: Ai challenges for testing resourcefulness and creativity. *Advances in Cognitive Systems* 6:31–44.

Sarathy, V. 2018. Real world problem-solving. *Frontiers in human neuroscience* 12:261.

Shen, W.-M., and Simon, H. A. 1989. Rule creation and rule learning through environmental exploration. In *IJCAI*, 675–680. Citeseer.

Shen, W.-M. 1989. *Learning from the environment based on percepts and actions*. Ph.D. Dissertation, Carnegie Mellon University.

Stefik, M. 1981. Planning with constraints (molgen: Part 1). *Artificial intelligence* 16(2):111–139.

Sussman, G. J. 1973. A computational model of skill acquisition. Technical report.

Yang, Q.; Wu, K.; and Jiang, Y. 2007. Learning action models from plan examples using weighted max-sat. *Artificial Intelligence* 171(2-3):107–143.

Ethics, Aesthetics and Computational Creativity

Daniel G. Brown
David R. Cheriton School of Computer Science
University of Waterloo
dan.brown@uwaterloo.ca

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

Abstract

We explore how the aesthetic lens of computational creativity can be used to aid in the development of ethical principles for artificial intelligence systems, and their application to real-world domains in which computers are expected to make reasoned, ethical judgments. In particular, we bridge two recent ICCC papers, one about how creative computers can design ethical principles, and one that uses algorithmic information theory as one component of the aesthetic value of the artifact. Our finding is that computational creativity ideas can enable the creation of novel ethical principles, but that the use of novelty, value and typicality measures in this space is quite challenging, and in particular, the algorithmic information theory objectives do not map smoothly to the goal of building fast ethical systems of provably high quality. We conclude with suggestions for making our approach usable in practice.

Introduction

AI systems, particularly those that inhabit the physical real world, make ethical decisions in response to either constructed dilemmas or ordinary scenarios all the time. This happens when they decide how to respond to human or other actors in need of help, but it also happens when a stock-picking robot decides which companies to invest in, or when an algorithm chooses which candidate to offer a job to, or (perhaps more) when the algorithm identifies which personal traits to look for in a successful candidate.

The all-pervasive nature of these ethical choices has caused "ethical AI" to become one of the current most active areas of research, teaching and progress in the area, with entire conferences devoted to engendering fairness (by multiple definitions), identifying properties of fair systems that are mutually incompatible, and reporting on situations in which an AI produces outcomes that are unfair, as when they make decisions that either confirm or exacerbate existing inequities, or when decisions are made by an AI for reasons that seem arbitrary. As such, concerns about training data bias, explainability, or the presence or absence of proxy variables that can be used to substitute for variables upon which discrimination is forbidden (such as height being a proxy for gender, or postal address as a proxy for race or income level) have also become major topics of research.

We argue in this paper that computational creativity (CC) has a message for AI ethics as well, but our focus is in fact that CC can produce ethical systems whose principles are themselves presented in a more aesthetic or satisfying way, and that the constrained exploration found in most CC systems can produce diverse systems of ethical principles, rather than reinforcing existing models for how robots or AI systems should interact with the world. Our argument bridges two recent ICCC papers: a paper by Ventura and Gates, which argues that considering AI systems as creative agents whose output artifacts are behavioral choices admits a natural approach for imposing ethics as an aesthetic filter on that behavior (2018); and the algorithmic information theory-based approach of Mondol and Brown, which seeks to use measures from that theory (basically, a few concepts from advanced Kolmogorov complexity) as indicia of high novelty and value (2021). A challenge with both of these papers is that they do not offer practical implementations.

Ventura and Gates consider the problem of ethical AI behavior on two levels. First, they propose a base-level system which considers potential behavioral choices and evaluates those choices via the lens of a normative ethics, which acts as an aesthetic and imposes ethical notions of novelty and value. They examine various classical normative ethical philosophies as possible behavioral aesthetics and conclude that the choice of *which* normative ethics should be used is a fraught one. In the most far-reaching part of their paper, they suggest building a meta-level creative system that creates abstract ethical principle sets and then focus on how to evaluate the ethical appropriateness of these meta-level artifacts, taking into consideration (meta-level) aesthetic notions such as novelty, utility, fairness, generalizability and comprehensibility.

Mondol and Brown also focus on quality and novelty, but their approach is much more abstract. For value, they indicate that an object is of high quality if it represents the output of a significant amount of computational effort (so called *logical depth*) or if it is an arbitrary member of a set described by a quite long program (also called *sophistication*). Objects with high logical depth are compressible; that is, they can be summarized with a very short program, but the program takes a lot of time to reconstruct the original object. Highly sophisticated objects are unusual, in that they show a large amount of internal structure, but describing that internal structure still requires substantially long descriptions; they are not just routine repetitions. Both of these measures are uncomputable. Another challenge is that if one is given a short, slow program that generates an object S, that alone is not proof that S is actually logically deep—another short program may also generate S, but do so in speedy run time, thereby demonstrating that S is in fact shallower than proposed. Similarly, just because there exists a long program for which S is a typical output (which might suggest that S is sophisticated) does not mean that there is not also a much shorter program that will also generate S as a typical output.

In the rest of this paper, we look into some of the properties of good ethical systems from the point of computational creativity, and explore ways in which Mondol and Brown's models of novelty and value can enter into the project of generating, comparing and evaluating ethical systems. We look into some of the contexts in which these systems might be used, and how creativity enters into the process of making ethical decisions as well. We explore how aesthetic judgments must be constrained here as well—just as a sonnet-generation system must create outputs that follow the rules and constraints of a proper sonnet, a system that builds ethical models must avoid horrifying outputs that may initially appear aesthetically pleasing.

Our approach is still theoretical—algorithmic information theory builds a collection of potentially theoretically sound, but fundamentally impractical, assessment tools for exploring the quality of objects like legal codes or computational systems, and designing algorithms to creatively explore the space of possible ethical systems will require a better sense of how to encapsulate both ethical systems and the dilemmas they face in a computational fashion. However, the combination of these two approaches offers the possibility of *incorporating aesthetics and ethics into a common framework, and developing a better understanding for how beauty and judgement can work together.* That said, there are key elements missing from "purely computational" frameworks—an aesthetics for ethics must also include a discussion of outcomes of ethical decisions, not just an analysis of their computational complexity and sophistication.

Two types of ethical system

Ethical decisions are made by computers or robots (or humans!) in response to specific situations. Here, we give two different formalisms for quite different scenarios in which agents make these decisions and describe how aesthetics can enter into the evaluation of this process. These two scenarios correspond to a large degree with the two levels of ethical CC agent treated by Ventura and Gates: the first considers the primary artifact to be a behavior (though we will see that this, in fact, may not actually be the locus of the creativity); the second considers the primary artifact to be analysis and judgement about behavior.

First, consider an agent residing in the real world. As a result of the state of that world, one piece of the process that decides what action the agent will take might be ethical—in response to a situation S, the ethical system P must quickly compute the best behavior b^* for the agent to perform. Or, P might generate a ranked list of behaviors $(b_1, b_2, \ldots, b_n)$, which information the agent uses in deciding what step to take next. In addition, each behavior may include a formal analysis A of why it makes a good choice. The key concern in this frame, though, is not interpretability; it is efficiency—for real-time decision-making, the system must compute $(b^*, A) = P(S)$ within a time bound t, or the decision will become moot. Nonetheless, for analysis of decisions that have been made by P, it is essential that its decisions are given along with some traceable analysis of where the decision b^* came from. Since P is fast, this can in theory be just a computation trace, which may be substantially unclear due to either deliberate or accidental obfuscation. Or, despite the fact that b^* must be computed quickly, it is possible that A may be computed reflectively and therefore much more slowly; indeed, many human justifications, both ethical and otherwise, may be computed this way as well (Bem 1972).

Whether or not A is interpretable and whether it is computed in real-time or post-hoc, it is arguably a much more significant output of P than is b^*, both from a creativity and from an ethical standpoint, especially if the set B of possible behaviors is well-defined and finite.[1] Nevertheless, in this instance, the agent can not be considered to be making deep ethical decisions, because it does not have time to do so; rather, it is quickly deciding how a (presumably) previously well-defined ethics P applies in situation S.

Second, consider the phenomenon of using legal codes to resolve disputes or trials. Here, there are two levels to the process: in the first, lawmakers must draft the law code C to be used as a basis for decisions. Traditional law codes, of course, are written in ambiguous natural language, and surrounding a code C will be existing jurisprudence and commentary C' upon which decisions can be easily hung.

Next, to respond to a dispute D, the judge must use reasoning based on the current law code C to produce an explainable outcome O for dispute D, such as a guilty verdict or a decision about financial damages (presumably from a well-defined and limited set $\mathcal{O}$ of possibilities), as well as a justification J for that outcome. As before, because $\mathcal{O}$ is (usually) a finite set, the justification J is the more creative task, as is building the way in which J comes from the law code and interpretations being used.

Both creative steps in this process are interesting from a computational creativity perspective: drafting C and drafting commentaries C' allows for one to explore questions of novelty and its appropriateness in a constrained space (we would not want to accidentally legalize murder as a "novel" innovation!), while at the same time, the choice of law code can enable simpler reasoning or force more complex reasoning in cases wherein the evidence of a particular case D is not well situated within the code C. As such, the "creativity" (particularly in the sense of novelty and value, but also in the sense of expressivity and conceptualization) of one choice of C or C' can have impacts on that of the other, and

[1] Classic ethical dilemmas, such as the well-known family of trolley problems, offer an agent two choices; which choice is made is never the point of proposing the dilemma, rather it is to elicit a justification for choosing that behavior (Thomson 2014).

they both have effects on O.

The difference between a law code C and commentary C' matters because law codes ought to be interpretable under a wide variety of situations, and in new contexts; for example, anti-discrimination law might not directly cite newly-protected groups, but expert commentary might offer arguments under an existing code that could be adapted to other groups than those previously identified.

We go into more detail about this mode below, but in this frame, an ethical decision process P is the creation (either in natural language, or in an interpretable format) of the pair $(O, J) = P(C, C', D)$.

Ethical decisions and quick actions

Many ethical dilemmas must be solved very quickly as part of an agent's participation in the world: should the agent interfere in an argument, call out a colleague for sexist behaviour, apply for an open job, choose one candidate for a job over another, and so on. So-called "trolley problems" (Thomson 2014), in which an agent must make a decision that will have negative consequences regardless of the choice made, also fit in this framework. These ethical dilemmas are encapsulated by a short description of the scenario and a straightforward decision that the agent has to make about the action it will take; perhaps this comes along with a predicted probability distribution over the states of the world that can result from the decision.

This formulation can easily turn into a Partially-observable Markov Decision Process (POMDP) if one simply optimizes expected long-term utility of the local decision at each step (Kaelbling, Littman, and Cassandra 1998). To avoid this potentially worrisome outcome, we note some ways in which this framing differs from the POMDP model.

First, the computational power of the agent may be restricted to the extent that solving a full POMDP may simply not be possible; at each step, there may be too many possible outcomes to choose from to do a full calculation of expected long-term utility, for example. Fundamentally, the decision maker operates in a position of bounded resources: it cannot fully model other actors in the situation, it may not have a proper description of the immediate probability distribution (let alone the long-term outcomes of choices) resulting from the decision it is making, and the limits on its computation may restrict the functions it can optimize. This is essentially the same argument used as one explanation for "non-rational" human behavior (Simon 1955).

Second, even the utility itself is not an easily-defined function. Instead, the agent itself will learn its utility function by assessing the outcomes of situations, both those that result from its own decisions, and those it is shown while it is being trained. As a result, it is even possible that this utility function will be time-dependent.

In this framework, computational creativity mostly enters into the design of the agent's ethical system itself and the assessment of its qualities. In particular, we look for aesthetic qualities in the way in which the agent responds to situations (both those found in training data and those found in its own experiences): can the agent's decision-making be said to model principles, can it be summarized in a way that generalizes from pre-existing data, and can it be expressed in a compact and easily computed way? A high-quality system should also be unaffected by irrelevant changes in the input, which in fact will allow it to operate with more efficiency. We also look to novelty: does the summarization algorithm function differently from previous algorithms despite generalizing the same data? One way to see this, consistent with Mondol and Brown, is to say that the algorithm derived to do fast ethical decision-making is not "typical" of existing algorithms of that sort—knowing *how* those algorithms work will not offer much assistance in compressing the description of a new ethical decision-making approach.

These aesthetic principles of parsimony, generalizability, consistency and (perhaps to a somewhat lesser extent) novelty are what we view as core ideas of a speedy ethical system. Can they be adapted to an algorithmic information theory model of value?

Legal decisions

Law codes have existed for millennia. Historically, they began largely as criminal codes, identifying behaviours not permitted for residents of a city or nation and the consequences thereof; over time, they expanded to much larger codes accommodating trade issues, family law and so on. At various times, magistrates and judges are given the task of applying existing legal codes to evidence coming from specific cases; they must build arguments based on the case, the law codes, and previous cases and their commentaries. Having humans do this work requires huge effort and expense: legal scholars must be trained to build and present arguments, and judges must use enormous law libraries full of precedents and commentary to adapt their jurisprudence to the situations of a contemporary dilemma.

We view this process as a set of creative tasks. In fact, from an aesthetic point of view, there are three quite different tasks that occur when one uses a law code to resolve a case. The first is the codification of the relevant law code itself—summarizing a collection of arguments, traditions and customs into a short natural language formulation.

The second aesthetic task is perhaps less obvious: how to present the information of a case. If the case is presented in a way that is true, but which obscures the way in which the law attaches to the details of the case, it can require much argumentation to be built in order to properly describe the decision (O, J) that best resolves the dilemma.

And the third aesthetic task is the one that is perhaps most interesting, and which can be assessed in a variety of ways: the process by which a judge (computational or human or a combination of the two) can build from a legal code C and commentary system C', and from the evidence of a case D to an outcome O with a justification J. If judgment is to be made by a computer, then the task is in a sense one of using existing arguments from C', together with rules from C, applied to case D to create the decision pair (O, J). If (O, J) is easily derived from the evidence and the legal information, then we can say that the bulk of the effort in the case was already done in the creation of those processes (and in the training of the computational judge). If, instead, much hair-splitting must be done in the task of interpreting the evidence of the case, then we can say that the code was not well-matched to the evidence.

This offers one of our most tantalizing realizations: namely, that the computational task of coming up with judgments can be seen as finding an efficient function that maps evidence D onto judgments J by filling in details from J quickly. If the decision J is easily created from D, given (C, C') as a legal code and advice, then (C, C', D) is a good set of evidence and laws for the case. In particular, we can say that knowing C' can help us resolve the case more straightforwardly.

To make this more formal, consider a collection of evidence D. Suppose there is a small set of possible outcomes $\mathcal{O}$ defined by the legal code C for cases of the type of D. In order to resolve the case, we must come up with the $O \in \mathcal{O}$ that best represents how D interacts with (C, C'), and the explanation J that requires the least extra computation on top of what has already happened in the creation of (C, C').

Creating an ethical decision process, then, consists of choosing a good decision-maker P, but also "priming the pump" by ensuring that P is well adapted to the law code C; in particular, P should be able to come up with verdicts for most cases quickly, and the pair (O, J) should be easily computed (at least for most cases) given the input data (C, C', D). In the language of Kolmogorov complexity, this corresponds to saying that the conditional Kolmogorov complexity of the decision (O, J) is small, given (C, C', D).

In particular, we note that a legal code that requires us to build long, involved judgments for simple cases, or for which small changes to the evidence could force us into a completely different set of valid justifications, is not a good one. Rather, for most cases, the mapping from D to the pair (O, J) needs to be efficient; that is, the legal code is preprimed to make fast, straightforward decisions.

Novelty and value in the context of ethics

Adapting traditional creativity measures to ethical systems and their products is a challenge. In particular, one principle that might be considered desirable in an ethical system is respect for precedent and tradition, which pushes these systems in a direction that moves away from novelty. Obviously, we still can look for new ways of reconsidering ethical dilemmas (either new ones or pre-existing ones), in the service of discovering a better way of improving people's lives, or in terms of mutually explaining a large number of compatible decisions. In this sense, the novelty of an ethical system is about the arguments it generates. As to value, the quality of an ethical system depends not only on the ostensible beauty of its philosophical tenets but also on objective observers' agreement with the decisions the system makes.

And of course, for scenarios in which the output of an ethical system is an argument or a legal code or a text decision, one can look at the overall quality of the text drafting, but of course, there is no value in a beautiful argument that creates a monstrous conclusion. In this sense, creativity may not always serve good outcomes, as when terrorists design novel forms of sabotage (Cropley, Kaufman, and Cropley 2008).

We can also look for some quality measures that are similar to those used by Mondol and Brown, which seek to encapsulate a collection of compatible ideas in a highly-compressible form with little internal redundancy. If generalizing these ideas can be done with a lot of effort, resulting in a short program that compresses the initial representation well, then this can be another indication of the value of the ethical system. Obviously, arguing about brevity alone is insufficient—an ethical system of the "kill all who oppose us" variety is clearly not a wise one despite its simplicity; rather, it is clear that wise ethics requires evidence of nontrivial thought from humans, or for computers, evidence of substantial computation.

Complexity-theoretic notions of aesthetic

Here we give a short introduction to the algorithmic information theory concepts Mondol and Brown use in their study of aesthetics. They use both *sophistication* and *logical depth* as measures of quality; we here focus on the simpler of these, which is logical depth. We also briefly summarize their approaches to novelty and typicality in non-technical language.

Basic Kolmogorov complexity concepts

A string s over a finite alphabet has Kolmogorov complexity $K_U(s)$ when this quantity is the length of the shortest input to a universal Turing machine U upon which U halts with the string s on its output tape. When U represents a programming language, $K_U(s)$ is the length of the shortest program in that language whose output is s; normally, we ignore the universal machine U and just speak of $K(s)$. There are a number of details about U that are also necessary (such as its accepting only a prefix-free language); we refer the reader to Li and Vitányi (2019) for full details.

The quantity $K(s)$ is uncomputable. In reasonable programming languages U, $K_U(s) \le |s| + c$ for some constant c, since one can just write a program that prints out the symbols of s one-by-one. In general, $K(s)$ represents an optimal compression of the information found in s. The value of $K(s)$ is not correlated with the usefulness of s. A random string has $K(s) \approx |s|$, which is high. The string 1^n of n consecutive 1's has $K(s) \le \log n + c$, since we can just write down the binary representation of the value n and then spit out that many 1s; this is a very low value of $K(s)$. (Certain values of n can be compressed far smaller than $\log n$; for these strings, $K(s) \ll \log n$.) And a string s of k random bits followed by $n-k$ 1's will have $K(s) \approx k + \log(n-k)$, which can take any value between $\log n$ and n. Knowing (just) the Kolmogorov complexity gives no way of distinguishing "useful" strings or "creative" strings from others.

Logical depth as value

Instead, Mondol and Brown move to estimate the value of a string s by its logical depth (Bennett 1988), the run time needed by short programs that compute s. Specifically, given a slip constant c,

$$d_{U,c}(s) = \min_{P:U(P)=s, |P| \le K(s)+c} \mathrm{time}(U(P))$$

that is, it is the minimum runtime of a program which generates s and whose length is within c of $K(s)$; again, both

U and c are often elided when they do not make the situation clearer. For simple strings, like those mentioned in the previous paragraph, $d(s) = O(|s|)$, because a PRINT program—in the case of the random string—and a linear-time FOR loop—in the case of the repeated symbol—will suffice to generate such strings (a simple combination of the two approaches suffices for the combination string). By contrast, a string s that contains the first n bits of a numerical constant that is hard to compute may be produced by a program whose length is a constant (plus a representation of the value n) but which takes a very long time to run; these are the logically deep strings. A short, slow program P whose output is a logically deep string s compresses that string very well, but an outside observer who does not have all the time needed for P to run will not be able to verify that it has a short program even if P is provided.

The overwhelming majority of strings are not even compressible, let alone logically deep (Li and Vitányi 2019). Mondol and Brown offer logical depth as one piece of evidence of the aesthetic value of a string; they propose that if a string is the output of a substantial, interesting piece of computation (or thought), then it is inherently valuable. One other component of this thesis is that as the length of s gets substantial, its availability to be compressed also grows; in particular, if s is the first n bits of a hard-to-produce constant, but the short, slow programs to produce that constant are longer than n bits long, then s is not logically deep—its shortest representation might in fact just be the program $\mathtt{PRINT}_s$. As such, logical depth is only meaningful as a function of long strings. By contrast, for long strings that come from repeated samples from a logically-deep creator, as the supply of these samples grows, the potential for finding repeated patterns and structures in those samples increases, and thus so may the possibility of actually finding a good compression method for such strings, including one that might require more complex algorithms than just “repeat this pattern k times”. Logical depth is a *property* of the string s, but the evidence for it is the short, slow program that generates the string. Given such a program P, we can confirm that the string is deep by running all programs of length at most $|P|$ for the time that P takes to generate s to see if any of them can produce s in less time, but this is impractical (indeed, so might be running P itself).

Using logical depth as a proxy for the value of an object raises a number of concerns, not the least of which is that it does not constrain the object to be properly a member of the class it needs to belong to; the text of a book might be logically deep, but if it is a brilliant mathematical text written in German, it is still not a high-quality English novel. Part of our goal with this paper is to consider this question of constraints—if suitably constrained by precedent, genre and custom, can logical depth serve as a proxy for value? A logically-deep legal code summarizes a large collection of cases in a very tight package, and the added information and computation needed to resolve dilemmas can be small; by contrast, an arbitrary legal code will either be trivial because it is simple, or trivial because it is random.

Conditional Kolmogorov complexity as novelty

Kolmogorov complexity also offers the possibility of identifying whether a new creative product is truly novel: an object is novel if knowing other members of its class offers little information about the new object. To make this formal, the *conditional Kolmogorov complexity* of s given t, $K(s|t)$, is the minimum length of a program which, when given t on its input tape, generates s on its output tape and halts. If $s = t$, the program just copies its input tape to its output tape, so the program is of constant length; if s and t are unrelated, then the program just ignores t, and $K(s|t) = K(s)$. A simple generalization allows the identification of $K(s|T)$, where T is a set of objects. Of course, conditional Kolmogorov complexity is just as uncomputable as ordinary Kolmogorov complexity.

Given a collection of objects $T = \{t_1, \ldots, t_n\}$, Mondol and Brown argue that if $K(s) \approx K(s|T)$, then s is *novel* with respect to T: the items in T do not help in describing s. Of course, this idea of novelty will be represented as a spectrum; for example, in practice, any English text will help to some degree in compressing any other English text, even if they are not from the same genre at all. Ens and Pasquier (2018) and Takamoto et al. (2016), among other authors, have used this measure to cluster items and identify their style, using general compressors to approximate conditional and absolute Kolmogorov complexity.

Models as typicality

One could use the opposite of novelty to model *typicality*, but Mondol and Brown instead use the concept of a model: given a set $T = \{t_1, \ldots, t_n\}$ of objects, we can build a program P_T, which, when run on given inputs $\{r_1, \ldots, r_n\}$ generates the items of T, with $P_T(r_i) = t_i$ for all i. This is called a *model* of T. Models are a restricted class of Turing machines; one variety of restrictions requires T to be a computable set and P_T to be a Turing machine that halts on all inputs.

If the model is a good one, then for all i, $|P| + |r_i| \approx K(t_i)$, and the members of T are considered *typical* for P. A new object s is also a typical member of the class if there exists a good model Q of $T \cup \{s\}$ such that $|P| \approx |Q|$; that is, learning about the existence of s does not make us have to do much more to accommodate it. A simple example of this phenomenon is that the program $\mathtt{PRINT}()$, which on input r prints r, is a good model for random strings, but a highly repetitive string s would not be “typical” for that class, as $\mathtt{PRINT}()$ is a bad model for such strings, since $K(s) \ll |s| + c$. In algorithmic information theory, this framing may also give a probability distribution over members of the class of outputs of P (Li and Vitányi 2019), and can be used to model properties of the overall class, assuming one has a good model.

Domain-agnostic vs. domain-specific aesthetic

The complexity-theoretic aesthetic measures proposed by Mondol and Brown are *domain-agnostic*. That is, they are concerned with abstract notions of complexity that are independent of the domain to which they are applied, and thus they can in principle be applied to any domain—one can imagine encoding a joke, a recipe, a piece of music, a mathematical theorem, a drug design or a legal code as a string and then using logical depth as an abstract measure of its value. However, as elegant as this is, it clearly does not capture everything there is say about value, when it comes to jokes, recipes, music, theorems, drug design and legal codes. In particular, it does not consider *domain-specific* notions of aesthetic, which do not generalize across domains—jokes should be funny, recipes delicious, music catchy, theorems influential, drug designs effective and legal codes fair.

While there may be general debate about whether creativity is itself domain-agnostic or domain-specific, we argue that it is both, at least as far as aesthetics is concerned.[2] This means that it is critical to determine how to integrate the domain-agnostic with the domain-specific for a unified theory of aesthetic—how do we ground abstract notions of complexity in a specific domain? Specifically here, how do we do so in the context of ethical decision making? One way to think about this is that the domain-specific aesthetics naturally constrain the space of possibilities (it may not be acceptable to choose murder as a conflict resolution, no matter how sophisticated the argument supporting it); within that constrained space, domain-agnostic aesthetics can be used to drive the search. Another paradigm may be that of multi-objective optimization, in which an agent attempts to satisfy (or satisfice) both the domain-agnostic and the domain-specific aesthetic measures.

Complexity-theoretic-based aesthetic in ethics

There are significant challenges with using the Mondol and Brown framework for identifying the quality of a creative artifact. First, and perhaps most distressingly, all measures used in their paper are uncomputable. Moreover, while their novelty metrics are largely just based on conditional Kolmogorov complexity between an object and others from an inspiring set, and can at least be estimated using standard compression algorithms like Lempel-Ziv (Ens and Pasquier 2018), the measures they identify for estimating the value of an object s largely relate to the internal complexity of that object; the only evidence of logical depth or sophistication is the creation of a slow short program whose output is s or a large model that generates s (as well as other objects) as a typical output.

As such, using computational complexity in any aesthetic scenario presents serious difficulties. However, this key challenge, ironically, is one of the strongest arguments in favour of the approach in the ethical domain: it recovers a fallacy often found in real human reasoning.

Charlatans and seemingly random decisions

A real annoyance, both in the real world and in computational artifacts, is claims that an object is serious and significant, when in fact it is arbitrary or random or trivial. This "The Emperor Has No Clothes" phenomenon is a serious risk of Mondol and Brown's formulation of value as sophistication or logical depth. For example, if P is a short program that first churns for $2^{|P|}$ useless steps, and then runs a very fast, very short, program P' whose output is a string x, then x will appear to be of high logical depth if we do not know about the program P'. Because in general it is impossible to know about the effects of a program without running it, programs of this sort are undetectable; indeed, as with the classic parable of the pot roast (in which a cook cuts off the ends of a beef roast before baking it for no reason other than that their parent did the same thing for a pan too small to hold a full roast) (Brunvand 1990), useless work might well be done by a contemporary reasoner because it arose in a benign former context and has never been discarded.

In our ethics framework, when the Emperor has no clothes, one of the objects under study for its aesthetic significance is assessed as having high logical depth or sophistication, by virtue of the long amount of research, study and preparation that has gone into its creation. But if that time has been wasted (by building circular logic, or by producing endless rehashing of the same case, for example, or by simply running a slow algorithm when a fast one might exist), the legal code C, or the decision outcome (O, J) may appear to be deep while not in fact being deep. (We note that detecting this scenario is difficult. For example, imagine if our standard for whether a string is logically deep or not is connected to polynomial runtimes. Then, if $\mathtt{P} = \mathtt{NP}$, there exists a fast compression algorithm for the binary string s_n that indexes graphs in a natural order $G_1, G_2, \ldots, G_n$ and has a 1 in position i iff graph G_i is Hamiltonian, which means that s_n is not logically deep; however, if $\mathtt{P} \neq \mathtt{NP}$, then no such fast compression algorithm exists, and s_n is logically deep.)

A different version of this problem occurs when the object under study was developed by a deliberately misleading agent. Here, the legal code C *appears* to be logically deep or of high sophistication: for example, we might even be able to run the short, slow program and create C with it. Such a program may still engage in some useless reasoning along the way of forming C, inserted by a charlatan who wants to make the code appear more serious than it actually is. Unfortunately, since in general it is hard (or uncomputable) to examine code for shorter or more efficient equivalents, it is also likely difficult to detect whether we have been deceived by a system that appears more complex than it actually is.

A similar problem arises when an extraordinary amount of detailed effort goes into planning how a system will respond to improbable scenarios. The object is *legitimately* logically deep and offers detailed guidance for how to handle the rare situation, summarizing challenging reasoning in a short package. Unfortunately, despite this potentially significant piece of work having been done, the author has hung it on a useless hanger. This situation is perhaps analogous to theological reasoning about the number of angels

[2] We hypothesize that this principle applies to creative process as well. That is, we hypothesize that there exists an abstract "core" creativity algorithm that is domain-agnostic and that can be specialized in domain-specific ways, rather like the notion of inheritance in object-oriented programming. However, we do not present arguments supporting that position here.

that can dance on the head of a pin—if this never observably happens, the system of reasoning is, in the domain-agnostic sense of Kolmogorov complexity, beautiful, yet useless.

Elegant is different than good

In addition to the concerns about seemingly random decisions, nothing stops an ethical system from being fundamentally monstrous except for external constraints pushing the decisions of that ethical system away from those terrible outcomes. In the previous subsection, we considered the case where a system appears sophisticated or logically deep, but is in fact not. However, one can also deploy algorithmic-information theoretic ethics in ways that are logically deep, but where the logical depth yields unhelpful results. For example, imagine a procedure P designed to decide cases about slavery, and outcomes of disputes involving enslaved people. If P is trained on a collection of cases and laws that start out with the presumption that slavery is valid, it might develop into a highly compressed program that encapsulates those cases and laws in an efficient framework. It might even use sophisticated reasoning to assert that one set of slaves should be freed and another subject to further bondage, generalizing from data about their cases and about existing similar cases. As such, P could appear to be typical and of high quality.

Further, P might not be like existing ethical systems in how it works, indicating that it is also of high novelty, in that knowing P does not given much help in building other pre-existing legal interpretation systems. However, none of these metrics—novelty, value, or typicality—pushes P to question the overarching unacceptability of the frame in which it operates. That is, P may be able to simplify, codify, and regularize the cases it decides, but if it starts with the requirement that it maintain the status quo, it may simply build a better evil system. It is unsurprising that this danger exists—it exists precisely due to the dichotomy of domain-agnostic vs. domain-specific notions of aesthetic.

Small changes with big differences

Another unexpected problem with the domain-agnostic measures of value and novelty is that they can push the system to make tiny changes to its texts that may have dramatic overall impacts. For example, suppose that C is a criminal code that identifies the sentences for violating various laws; for simplicity, suppose that C_1 is a function that maps a crime c to *days* in jail $C_1(c)$. The code C_1 is essentially equivalent in complexity to another code C_2 that assigns the same number of *weeks* in jail to c as C_1 assigns days. (That is, $C_2(c) = 7C_1(c)$ for all c.) Yet these are fundamentally different. Similarly, and more alarmingly, imagine that C_1 is a civil code that describes how to identify which party legally owns a piece of property under dispute between two parties. If C_2 is a new civil code that results in exactly the reverse outcomes to that of C_1, then both C_1 and C_2 are essentially equal in all measures of complexity, just as a photograph and its inverse are.

The only way to avoid this problem is via precedent—we must prime the pump with existing case law, and only accept legal codes that are consistent with existing decisions. But this leaves us in the position we were hoping to avoid—novelty comes not through generalizing from existing situations, but by intentionally moving away from what is known.

Not all bad news

This litany of negative news about Kolmogorov complexity-based aesthetic might suggest that the whole endeavour is hopeless, but that is not the case. The fundamental idea still appears sound: a short legal code, or a simple, fast ethical system, which summarizes a large amount of case law in a small, efficiently-presented package, and which allows for the fast resolution of simple cases and a complex reasoning process in difficult cases, is exactly what is needed.

To be more specific, consider a legal question (C, C', D). If D is easily resolved, it should be the case that $K((O, J)|(C, C', D))$ should be small—that is, it should be possible to efficiently fill in the details of a proper judgment given the legal code and commentary, with very little extra information. Creating this extra information is, ultimately, the task of the judge, and the key observation is that if D is a *typical* case for (C, C'), this work of finding a good resolution for it should be efficient. By contrast, if D is an odd edge case, then the judge must perform substantial computation, creating much new information, in computing the outcome (O, J) of the dispute.

Fundamentally, then, an aesthetically appealing ethical system, particularly in our second frame, consists of a concise representation of complex ethical principles with an algorithm for quickly mapping them onto resolutions for dilemmas that commonly arise. Further, novelty search should enable the discovery of both algorithms and principles that, while they encapsulate similar information to those pre-existing, nonetheless use different methods; that is, knowing about existing algorithms should offer minimal capacity to predict a new approach.

Building an ethical system

As in Ventura and Gates, now comes the rub: how do we develop a system whose output is novel, valuable, consistent, transparent, and non-trivial? In no small part because of the challenges described in the previous section, we largely leave this question for future work and analysis.

As one possible avenue of exploration, as briefly suggested by Ventura and Gates, it may be possible to perform large-scale statistical simulations involving agents making decisions using the ethical system under scrutiny. Serendipitously, this is possible exactly because the agents are computational rather than human, and, interestingly, this empirical approach could apply to estimating both the domain-agnostic, information-theoretic aspects of aesthetic as well as the domain-specific, ethics-based aspects. For the former, one may be able to use statistical simulations to estimate information-theoretic measures similar to how Soler-Toscano et al. empirically estimate the algorithmic probability of short strings (2014). For the latter, such simulations may be used to estimate the likelihood of various individual outcomes, to perform differential analysis, or to model large-scale social outcomes, facilitating a comprehensive/empirical description of the system in terms of its effects.

For example, considering the case of real-time ethical decision making, we might construct a simulation of self-driving vehicles encountering ethically challenging scenarios. [3] Driving agents could be equipped with varying ethics systems and large-scale simulations could result in statistical measures of global or local utility (e.g., to estimate fairness). Or, agent behavior patterns could be analyzed for complexity as in (Zenil, Marshall, and Tegnér 2015) (e.g., to estimate logical depth).

For the case of making legal decisions, many of the same kinds of ideas might be applied. For example, a system for drafting traffic laws might hypothesize a traffic code and then perform simulations of varying traffic scenarios governed by that code to verify its generality, its clarity or its fairness statistically. Or, perhaps the complexity of the simulation may act as a proxy for the complexity of the code, in the information-theoretic sense.

The the main difference between the two scenarios is the perspective from which simulations are being run and who is making use of them for what: in the first case, if we are using simulations to evaluate something about the ethical system, it is because we are designing the system and wonder about its utility for the agent; in the second case, the agent itself is constructing the ethical system and is using the simulations as a part of its creative process.

Aesthetics of creating ethical systems

We have identified the creation of ethical systems as a fundamentally creative task, and considered the aesthetics of this task under two quite different formulations: building fast algorithms that find solutions to ethical dilemmas (as well as explanations for those solutions), and building slow algorithms that reason using law codes to find the correct answer to a serious case, and offer detailed reasoning to justify their decisions. We have suggested that aesthetic judgments appear at multiple steps in this process, and in particular, that good design of legal codes can enable efficient decision making and more transparent reasoning.

We also briefly discussed the actual process of searching for such ethical principles. A key issue is that they must not be assessed solely on the basis of novelty, typicality and value as measured by (domain-agnostic) complexity, but (domain specific) characteristics such as fairness and real-world suitability must also be considered; failing to account for the latter creates the possibility of developing ostensibly beautiful philosophical models that are monstrous in the real world. While complexity-theoretic-based aesthetics can play a role in the development of ethical systems, these systems must still generalize from the decisions of extant judgment systems and case law, and they must display straightforward properties (such as consistency, explainability and generalizability) that are found in real-world systems.

Interestingly, this multiple-step process of looking for ethical answers, and then looking for ethical systems, suggests that we could also go out one further level, to the aesthetic analysis of the procedure with which we search for ethical systems. That is, we can have an aesthetics of ethical decisions and an aesthetics of ethical systems, but we can also assess the aesthetic value of the process of building systems that build ethical systems. Much as with the existing dilemmas of this paper, incorporating novelty, value, typicality and feasibility into such an assessment will likely not be an easy task.

Conclusions

In this work, we have looked at the question of ethical decision making and the design of ethical frameworks, to see if computational creativity offers different advice for this process than do other existing frameworks. We used the frame of Ventura and Gates, who first proposed this aesthetic understanding of the process of finding good ethical principles, and combined it with the domain-agnostic aesthetic value measures of Mondol and Brown, which focus on conditional computational complexity and efficiency of summarization as measures of novelty and value. We argue that we might use such an approach to aesthetics on both the process of making ethical decisions and the process of designing ethical systems, but in practice the challenges with computing these measures, and the potential that a decision maker might build something ostensibly aesthetically beautiful, but in practice monstrous, still remain. Putting this whole approach into practice will require much further work.

We note, finally, that while our motivation for considering these questions has been the question of developing ethical systems and the computational creativity of this question, there is ultimately nothing fundamentally ethics-based about many of our arguments; the same types of arguments likely hold for developing computationally creative systems that parent children or train pets, that make theological arguments, or that otherwise generalize reasoning from a large case set, or make quick decisions. We look forward to both generalizing these results and finding ways to make them more practical.

Acknowledgements

The research of D.B. is supported by an NSERC Discovery Grant.

Author Contributions

Both authors contributed to all aspects of the work, including ideation, narrative/position development and writing.

References

Bem, D. J. 1972. Self-perception theory. In Berkowitz, L., ed., *Advances in Experimental Social Psychology*, volume 6. New York: Academic Press. 1–62.

Bennett, C. H. 1988. Logical depth and physical complexity. In Herken, R., ed., *The Universal Turing Machine: a Half-Century Survey*. Oxford University Press. 227–257.

Brunvand, J. H. 1990. *Curses! Broiled Again!* W.W. Norton.

[3] For example, something like this: `https://www.moralmachine.net`

Cropley, D. H.; Kaufman, J. C.; and Cropley, A. J. 2008. Malevolent creativity: A functional model of creativity in terrorism and crime. *Creativity Research Journal* 20(2):105–115.

Ens, J., and Pasquier, P. 2018. CAEMSI : A cross-domain analytic evaluation methodology for style imitation. In *Proceedings of the Ninth International Conference on Computational Creativity, Salamanca, Spain, June 25-29, 2018*, 64–71.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101(1):99–134.

Li, M., and Vitányi, P. M. 2019. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer Publishing Company, Incorporated, 4th edition.

Mondol, T., and Brown, D. 2021. Incorporating algorithmic information theory into fundamental concepts of computational creativity. In *Proceedings of the International Conference on Computational Creativity*, 173–181.

Simon, H. A. 1955. A behavioral model of rational choice. *The Quarterly Journal of Economics* 69(1):99–118.

Soler-Toscano, F.; Zenil, H.; Delahaye, J. P.; and Gauvrit, N. 2014. Calculating kolmogorov complexity from the output frequency distributions of small turing machines. *PLoS ONE* 9(5):e96223.

Takamoto, A.; Umemura, M.; Yoshida, M.; and Umemura, K. 2016. Improving compression based dissimilarity measure for music score analysis. In *2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA)*, 1–5.

Thomson, J. J. 2014. Killing, Letting Die, and The Trolley Problem. *The Monist* 59(2):204–217.

Ventura, D., and Gates, D. 2018. Ethics as aesthetic: a computational creativity approach to ethical behavior. In *Proceedings of the International Conference on Computational Creativity*, 185–191.

Zenil, H.; Marshall, J. A.; and Tegnér, J. 2015. Approximations of algorithmic and structural complexity validate cognitive-behavioural experimental results. `https://arxiv.org/abs/1509.06338`.

Should we pursue SOTA in Computational Creativity?

Anna Jordanous
School of Computing
University of Kent
Canterbury, Kent, UK
a.k.jordanous@kent.ac.uk

Abstract

Should we pursue a *state-of-the-art* in Computational Creativity? The activity of 'SOTA-chasing', or working towards beating performance standards achieved by the current state of the art, is typical in many research disciplines relevant to computational creativity such as Machine Learning or Natural Language Generation (SOTA). Computational Creativity (CC) research does not typically engage with SOTA-type benchmarks. Consequently, it becomes harder to objectively identify high-performing systems in a creative domain (area of creative application), despite our research efforts building significant bodies of work in several domains. This paper critically engages with the use of SOTA in other related disciplines and explores the idea of working with SOTA-based evaluation in CC. The paper offers recommendations for (careful) use of SOTA to invigorate and direct CC progress.

Introduction

Should we pursue a *state-of-the-art* in Computational Creativity? In many AI disciplines related to computational creativity, typical research practice includes some evaluation experiments to compare research results to a ground truth set of results derived from some comparable benchmark or leading system in the same research area, referred to as the current state-of-the-art (SOTA). In Computational Creativity, for various reasons, the idea of a SOTA has frequently been dismissed as irrelevant and/or unachievable, despite our research efforts building significant bodies of work in several domains (areas of creative application). The consequence is that it becomes harder to identify which are the leading systems in a creative domain, in terms of inspiration or in terms of representing the bar that has been set for achievements and knowledge advances in computational approaches to creativity in these domains.

SOTA and its use in AI research

SOTA stands for State Of The Art, and refers to some leading benchmark or system for a particular task. In many AI disciplines relevant to computational creativity, such as Machine Learning or Natural Language Generation, it is typical to perform at least some evaluation in comparison to a ground truth baseline or set of results derived from the current state-of-the-art (SOTA) for that research task. This has become standard practice, to the extent that the acronym SOTA has become a recognised noun in AI research vocabulary. SOTA is typically measured objectively, either numerically or as a percentage, via metrics that have come to be recognised as appropriate for that task. Common metrics include accuracy and specificity, statistical tests, or F-scores (a combinatory measure of precision and recall).

What has also become standard practice in such disciplines is the activity of 'SOTA-chasing', or trying to better the performance of the current state of the art. This is typically encouraged. The guidelines for the International Joint Conference in Artificial Intelligence (IJCAI), a leading AI conference,[1] refer its reviewers to guidance (Blockeel and Davis 2022) that asks reviewers to evaluate experiments in a paper based on various criteria such as "'Are competitors SOTA? Are all competitors chosen? If not, how have they been selected? Are the conclusions aligned with this selection? ... this information is relevant for assessing how convincing the experimental results are" (Blockeel and Davis 2022, slide 41).

Historical perceptions of SOTA in CC

Computational Creativity is not a discipline where we tend to record or measure any state-of-the-art. Within the field, objective evaluation metrics based on the product of a creative system such as Ritchie's empirical criteria (Ritchie 2007), once quite popular, are now not used very often. Such objective evaluation was criticised for only evaluating the product of creative systems, ignoring the process by which they operated (Colton 2008), and other of the Four Ps of creativity (Jordanous 2016) (Producer, Press, Product, Process). Ritchie's criteria also required some agreement on domain-appropriate choices of threshold values and parameters for the criteria. However we have seen Ritchie's criteria deployed with some success for comparative evaluation in the areas of narrative generation (Pereira et al. 2005) and music improvisation (Jordanous 2012).

Other generic evaluation metrics or frameworks exist such as FACE (Pease and Colton 2011) and the Creative Tripod (Colton 2008), or domain-specific evaluation metrics such as O'Donoghue's statistical tests for analogies (O'Donoghue

[1] https://ijcai-22.org/faqs/

2007). These tend to be implemented using subjective judgements, difficult to replicate consistently for comparison over time due to possible variability in human opinion.

SOTA: Meaningless numbers?

If we did try to deploy some sort of objective metric for evaluation in CC, what would the measurements actually represent? Wouldn't numeric measurements or percentages be meaningless? Not necessarily. Objective metrics have been proposed that could be used for comparative evaluation against some established baseline, such as the work by Bossou and Ackerman (2021), or the (to-date, unused) IDEA model (Pease and Colton 2011) and previous tests proposed by Pease, Winterstein, and Colton (2001). It is also not impossible to consider ways in which methods such as FACE and the Creative Tripod could be operationalised in objective metrics. The SPECS evaluation methodology (Jordanous 2012) also opens up ways for evaluative tests to be defined relative to a definition of creativity, which could be defined objectively. We have seen specific uses of evaluation metrics defined for particular domains (areas of creativity), such as the use of scores for story plots (Pérez y Pérez 2014).

Comparative evaluation as a blunt tool in CC?

What does it mean to measure or compare one system against each other? It seems unrealistic to pursue the devising of universal SOTA benchmarks that might cover all different types of creative systems. But that should not stop us in our tracks. Fields such as Machine Learning use SOTA benchmarks to compare applications or algorithms that work on roughly the same task, that can be directly compared.

Do we have enough effort in particular applications of creativity to have a meaningful domain-specific SOTA benchmark for that area? While we have seen arguments for (and evidence of) comparative evaluation being useful to measure progress (Jordanous 2012, e.g.), a common feeling in earlier days in CC was that it does not make sense to evaluate systems against each other, as we did not have enough comparable systems to establish a state of the art. CC has now reached a stage, however, where there are various application domains that are well represented in terms of different systems (Loughran and O'Neill 2017).

A more subjective objection might be that it feels to some extent inappropriate to have a system identified as best-performing in a specific domain of creativity, due to the wide variety of ways in which creative systems can excel even if performing comparable tasks. (We should acknowledge that this has not stopped the existence of human-equivalent competitions of the 'best' artist, or story-teller, or idea generator, for example, nor the monetary valuing of creative outputs.)

But without recognising the achievements of some systems as superior to others, how can we hope to learn from the systems that do outperform others? Let us consider the potential benefits of some kind of SOTA-based evaluation.

Potential benefits of SOTA evaluation

If we could use SOTA-based evaluation in CC, would the field benefit? In other words, if we could establish metrics that captured a state-of-the-art baseline in various domains that are well-covered by Computational Creativity research, such as narrative generation, visual art generation, or music composition, then what would we gain from testing new systems in those domains against the current state-of-the-art? Learning from other disciplines that use SOTA, we could have tangible ways to measure progress in particular research domains (Lewis and Crews 1985). This might help computational creativity research venues to establish greater credibility within more general AI conferences such as IJCAI, ECAI, AAAI and so on, where our typical papers may not currently be seen as containing enough rigour due to lack of comparative experiments against a SOTA. Perhaps more importantly, if we could establish SOTA for a particular CC domain, then this would be transferable to those working outside of the direct CC community. Looking at conferences in the remit of computational creativity such as ISMIR (music) or ACL (language), it is still possible to have papers accepted with 'hand-wavy' justifications of a system being considered creative with little or no rigorous evaluation of that claim of creativity; because (within my own subjective experience) there is little adoption of CC's creativity evaluation metrics outside of the CC field itself.

Does 'SOTA-chasing' give us a clearer idea of the best current systems in a particular area? And when a new system represents a significant advance? After all, our current ways of identifying the current state of the art are subjective, hence vulnerable to misinterpretation and bias.

There is of course significant pressure to get appropriate metrics of strong performances in a creative domain. Pursuing a SOTA benchmark for a domain could help us establish objective metrics for evaluation, available for reuse to compare systems (typically considered good practice in terms of establishing how systems represent advances in knowledge).

Potential risks of SOTA evaluation

Use of SOTA evaluation in AI/ML areas is common, and accompanying this is the risk of getting only minor incremental advances - where papers could be considered ready to publish if they advance SOTA by a minuscule percentage. At the other end of this extreme, we are a field which typically encourages innovation in method and approach even if there is not a tangible effect on results; we do not want to be in the situation where a system that does not beat SOTA becomes almost unpublishable.

'SOTA-chasing' as a research activity has been criticised by some (Church and Kordoni 2022; Koch et al. 2021). One criticism of particular relevance to CC is the question of what approach to take if we do not have a direct or obvious-fit metric to use. There is no one 'test for creativity'. In this circumstance, we can examine what another similar field does. Thanks to the likes of the GPT-* transformer systems et al, deep learning-based text generation has seen phenomenal progress over the past few years. Typically, such systems need to evaluate output at scale, with large data output to evaluate that needs automated metrics. Lacking a specific automatable metric for evaluating generated text (a problem familiar to those working with creative language generation), it is common to see the machine translation metric

BLEU used as a *proxy* for evaluating the success of a system in learning from some input data to generate some output data. In other words, such a metric is considered to be an approximate evaluation of success: 'good enough' to be adopted in order to facilitate progress.

What happens if we use the wrong metrics, or fail to evolve or adapt our metrics over time as needed? The reliability of experimental research depends heavily on how research findings are derived from scientific experiments (Ioannidis 2005). Does research go in the wrong direction? Taking our example of transformers research, only time will tell, but the phenomenal progress over the past few years seems to suggest that the adoption of a 'good enough' proxy metric has been helpful in allowing research to progress. In such situations, community self-awareness of the metric's status as a proxy, not a direct measurement, is critical.

Recommendations for use of SOTA in CC

Would SOTA chasing be enough to replace other evaluation? No, probably not, particularly as this would reverse progress in creativity evaluation and risk us forgetting what we have learned about evaluating our creative systems (see the *Historical Perceptions* discussion above). *But it could complement what we are already doing.*

We should acknowledge that even in disciplines where SOTA-based evaluation has come to be typical, it is not mandatory for research success and such research communities do not always advocate for experiments referencing and comparing to SOTA. Although, as remarked above, the IJCAI conference refers reviewers to recommendations (Blockeel and Davis 2022) to check if experiments compare a piece of work against SOTA, the same guidance also states what to do if you are reviewing a paper where there is:

> " "No experimental comparison to SOTA". Ask yourself: is it needed?
>
> - In 95% of cases: yes. But be aware of that 5%.
> - e.g.: theoretically very innovative work, novel insights, ... may be valuable even if an implementation or experimental comparison is not possible at this time"

(Blockeel and Davis 2022, slide 39)

The *Historical perceptions* section above reflects on how we could implement SOTA metrics in ways which do not focus just on measurable aspects of creative output, but which measure process and other Four P perspectives. In some contexts, a SOTA benchmark would be establishable with current metrics (fine-tuned towards some objective metric, as discussed above). In fact, it could be argued that this has already happened in the context of poetry evaluation (Pereira et al. 2005). We could delve into the statistical and empirical measurements and tests common in AI and data science, and see what could be used, as for example in O'Donoghue (2007). There are other measures of subjective concepts that we could learn from and perhaps re-appropriate for SOTA metrics, for example, Seth's measure of autonomy and emergency (Seth 2010).

Proposal: CC-eval competition

In research areas such as music Informatics, NLP, and multimedia computing, as well as (even closer to home for CC) procedural content generation, research progress is aided by evaluation against benchmarks, as part of regular (typically annual) competitions. See for example:

- MIREX (music) `https://www.music-ir.org/mirex/`
- SemEval (NLP) `https://semeval.github.io/`
- MediaEval.org (multimedia computing) `http://www.multimediaeval.org/`
- GDMC - Generative Design in Minecraft (PCG) `http://gendesignmc.engineering.nyu.edu/`
  - (and GDMC's Chronicle for games narratives)

Does CC need a CC-eval competition like MIREX, SemEval, and so on? We have in the past seen curated exhibitions in past ICCCs, so we have an established vehicle within which to host such an event each year. And let it be remembered that we do already do competitions, or at least those of us working in PCG do. The GDMC competition has seen considerable growth in the few years it has been operating, acting as a high visibility route into established and well-defined PCG challenges. Treating GDMC as a test case, it's important to recognise that the use of metrics based on human judgement requires a lot of effort on the part of judges. This has led to exciting work with GDMC organisers exploring automatable metrics (Hervé and Salge 2021)

How could a CC-eval competition work? This could follow a MIREX-like model of proposed tasks each year, many of which may re-occur from year to year. In this model, the task proposers also propose evaluation metrics that are applied to all entries (and any 'inspiring set'/training data).

Such a competition could provoke interest in pre-defined tasks (as GDMC and SemEval/MediaEval/MIREX do), with potential benefits of attracting new researchers and also keeping established researchers engaged (and challenged by the 'new kids on the block'!) Such competitions have seen their tasks form the basis of student projects at undergraduate level and above. They have been useful for community spirit building and the establishment of GroundTruth metrics by those working directly in a creative domain who feel confident enough to propose and run the task that year. Metrics could be examined and used every year that a task runs.

This proposal comes with downsides, of course. We would need to tackle many challenges outlined in this paper, particularly if proposing a task. Initial task metrics would require some very careful thinking, ideally crowdsourcing via experts in that CC domain. For subjective evaluation metrics, could we get enough commitment from judges? MIREX have in the past struggled with this, for example. There would be considerable obstacles in terms of set-up effort, time commitment, organisational infrastructure and reliance on volunteers, at a time when many of us are exhausted and burnt-out from pandemic related uncertainties and workloads. But perhaps this would help us come together to reinvigorate that part of our community spirit that is so hard to replicate if not meeting every year in person, as well as create an exciting entry point for newcomers?

Conclusions

The field of Computational Creativity has thus far resisted the idea of establishing and bettering a current state-of-the-art target for specific domains. SOTA-chasing has become the norm in various sub-fields of AI such as Machine Learning (ML) or Natural language Generation (NLG). As commented above, recent advances in NLG provide an example of the remarkable progress that can be facilitated through using SOTA benchmarks for targeted improvement, even when metrics are not as clearly identifiable as in tasks which can be measured using statistical or information-theoretic measures.

My argument in this paper is that meeting or beating SOTA in CC is not the requirement it is billed to be in ML, and it also is not the devil it could sometimes be perceived to be in CC. I suggest CC research has reached a point of maturity where we can start doing it, to help us track progress in each creative domain that we have built up a body of work in. This will help build the field, as long as we can learn from those in related disciplines and avoid weakening our research due to falling into the traps identified by Goodhart's law - "when a measure becomes a target, it ceases to be a good measure" (Oxford Reference retrieved May 2022).[2]

There are many pitfalls to be aware of. What I propose here should not replace more substantial evaluation, but could complement it. Pursuit of SOTA metrics could help us in the pursuit of evaluation metrics, as well as adding a new way to target and track progress and even help build our community further. I posed a possible route forward of a CC-Eval competition, as a *Grand Challenge* for CC, inspired by the likes of MIREX and SemEval (but I should stress this is one of many possible routes forward).

We should acknowledge that metrics for measuring SOTA in a creative domain may need to change over time, to avoid the criticism that credibility of a scientific field of research is weakened by lack of flexibility for that field to self-correct (Ioannidis 2012). As one reviewer of this paper commented, we also need to be familiar the meanings and intentions behind the metrics we use, to critically appreciate the levels of meaningfulness and informativeness of results.

Our research community (and domain sub-communities) contain enough domain expertise to recognise and collectively establish the most appropriate metrics for a creative application area. As a community, we have a history of engaging enthusiastically with self-reflection and self-correction (see for example the paper types in the Short Paper call for this conference). We also have a history of considering evaluation of creativity deeply, including metrics for meta-evaluation that we could apply to our tests for SOTA benchmarks (Jordanous 2014).

What we do need, to progress this further, is for people working in specific areas of computational creativity to propose, use, evolve and convalesce onto some SOTA metrics for those areas. These metrics do not need to be perfect; we know this is pretty much impossible in many creative domains. However careful choosing of 'good-enough' metrics as a proxy for that creative area - as the text generation community have done - opens doors for tracking and furthering progress in various domains of Computational Creativity.

Author Contributions

AJ ideated and wrote the paper alone, though as acknowledged below, the paper benefitted much from discussions.

Acknowledgments

The inspiration for this paper comes out of failed attempts to explain to my colleagues at the School of Computing, University of Kent, why CC does not pursue state-of-the-art comparisons. I also thank Piotr Sawicki, Marek Grzes, Dan Brown and Max Peeperkorn for our conversations around the use of SOTA. In particular, conversations with Max Peeperkorn prompted me to finally write up these ideas for ICCC'22, after my pondering on them for some time.

References

Blockeel, H., and Davis, J. 2022. A brief introduction to reviewing. https://dtai.cs.kuleuven.be/introduction-to-reviewing-for-ijcai.pdf , last retrieved 22 April 2022.

Bossou, K., and Ackerman, M. 2021. Should machines evaluate us? opportunities and challenges. In *Proceedings of the 12th International Conference on Computational Creativity, Mexico City, Mexico*.

Church, K. W., and Kordoni, V. 2022. Emerging Trends: SOTA-Chasing. *Natural Language Engineering* 28(2):249–269.

Colton, S. 2008. Creativity versus the Perception of Creativity in Computational Systems. In *Proceedings of AAAI Symposium on Creative Systems, Stanford, US*, 14–20.

Hervé, J. B., and Salge, C. 2021. Comparing PCG metrics with Human Evaluation in Minecraft Settlement Generation. *ACM International Conference Proceeding Series*.

Ioannidis, J. P. A. 2005. Why most published research findings are false. *PLoS Medicine* 2:e124.

Ioannidis, J. P. A. 2012. Why science is not necessarily self-correcting. *Perspectives on Psychological Science* 7(6):645–654.

Jordanous, A. 2012. A Standardised Procedure for Evaluating Creative Systems: Computational Creativity Evaluation Based on What it is to be Creative. *Cognitive Computation* 4(3):246–279.

Jordanous, A. 2014. Stepping back to progress forwards: Setting standards for meta-evaluation of computational creativity. In *Proceedings of the 5th International Conference on Computational Creativity, Ljubljana, Slovenia*. Ljubljana, Slovenia: ACC.

[2] The excellent comments from the anonymous reviewers, including the reference to Goodhart's law, demonstrate how CC researchers can - and do - engage very productively with this debate, even if one does not agree with the arguments I present here.

Jordanous, A. 2016. Four PPPPerspectives on computational creativity in theory and in practice. *Connection Science* 28(2):194—-216.

Koch, B.; Denton, E.; Hanna, A.; and Foster, J. G. 2021. Reduced, Reused and Recycled: The Life of a Dataset in Machine Learning Research. In *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS 2021), Online*.

Lewis, B. C., and Crews, A. E. 1985. The evolution of benchmarking as a computer performance evaluation technique. *MIS Quarterly* 9(1):7–16.

Loughran, R., and O'Neill, M. 2017. Application Domains Considered in Computational Creativity. In *Proceedings of the Eighth International Conference on Computational Creativity (ICCC'17), Atlanta, GA*.

O'Donoghue, D. P. 2007. Statistical evaluation of process-centric computational creativity. In *Proceedings of the 4th International Joint Workshop on Computational Creativity, London, UK*.

Oxford Reference. retrieved May 2022. Goodhart's law. https://www.oxfordreference.com/view/10.1093/oi/authority.20110803095859655.

Pease, A., and Colton, S. 2011. Computational Creativity Theory: Inspirations behind the FACE and the IDEA models. In *Proceedings of the 2nd International Conference on Computational Creativity, Mexico City, Mexico*, 72–77.

Pease, A.; Winterstein, D.; and Colton, S. 2001. Evaluating Machine Creativity. In *Proceedings of Workshop Program of ICCBR-Creative Systems: Approaches to Creativity in AI and Cognitive Science, Vancouver, Canada*, 129–137.

Pereira, F. C.; Mendes, M.; Gervás, P.; and Cardoso, A. 2005. Experiments with Assessment of Creative Systems: An Application of Ritchie's Criteria. In *Proceedings of the Workshop on Computational Creativity (IJCAI 05), Edinburgh, UK*.

Pérez y Pérez, R. 2014. The Three Layers Evaluation Model for Computer-Generated Plots. In *Proceedings of the Fifth International Conference on Computational Creativity, Ljubljana, Slovenia*.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17:76–99.

Seth, A. K. 2010. Measuring autonomy and emergence via Granger causality. *Artificial Life* 16(2):179–196.

Putting GPT-3's Creativity to the (Alternative Uses) Test

Claire Stevenson, Iris Smal, Matthijs Baas, Raoul Grasman & Han van der Maas
Psychological Methods Department
University of Amsterdam
Roetersstraat 11, 1018 WB Amsterdam, The Netherlands
c.e.stevenson@uva.nl

Abstract

AI large language models have (co-)produced amazing written works from newspaper articles to novels and poetry. These works meet the standards of the standard definition of creativity: being original and useful, and sometimes even the additional element of surprise. But can a large language model designed to predict the next text fragment provide creative, out-of-the-box, responses that still solve the problem at hand? We put Open AI's generative natural language model, GPT-3, to the test. Can it provide creative solutions to one of the most commonly used tests in creativity research? We assessed GPT-3's creativity on Guilford's Alternative Uses Test (AUT) and compared its performance to previously collected human responses on expert ratings of originality, usefulness and surprise of responses, flexibility of each set of ideas as well as an automated method to measure creativity based on the semantic distance between a response and the AUT object in question. Our results show that -on the whole- humans currently outperform GPT-3 when it comes to creative output. But, we believe it is only a matter of time before GPT-3 catches up on this particular task. We discuss what this work reveals about human and AI creativity, creativity testing and our definition of creativity.

Introduction

A New York Times magazine headline (April, 2022) states "OpenAI's GPT-3 and other neural nets can now write original prose with mind-boggling fluency…". Reactions to this on Twitter and in blogposts vary, but many converge on the sobering belief that large language models (LLMs) are somewhat of a hype and err on the side of 'stochastic parrots', a reference to the computational linguists Bender et al. (2021) comments on the uses and dangers of large language models. We could easily take this a stretch further and argue LLMs have not achieved general artificial intelligence (Mitchell, 2021; van der Maas, Snoek and Stevenson, 2021), much less "truly" creative artificial creativity.

In daily life, such LLMs, and GPT-3 in particular, have proved very useful in (co-)creating phenomenal works: newspaper articles (e.g., GPT-3, 2020), novels (e.g., Green, 2020), and poetry (e.g., Aalho, 2021). The first author even has students who have admitted to using LLMs to help write their theses. These works meet the criteria of the standard definition of creativity: being original and useful, and sometimes even the additional element of surprise (Runco and Jaeger, 2012). But, how much creative thinking can be attributed to such LLMs? Can such large language models really produce the creative insights that humans are capable of?

In this study we put one particular LLM's creativity to the test, OpenAI's GPT-3 (Brown et al., 2020). We compare its performance to that of humans on the popular Alternative Uses Test (AUT, Guilford, 1967). On the AUT people are asked to produce as many original uses for an everyday object as possible (e.g., a 'brick' can be used as a 'paperweight' or 'to break a window'). Responses to the AUT are generally scored in terms of quality, such as the originality and utility of each generated idea, often rated by two or more experts using the Consensual Assessment Technique (Baer and McKool, 2009). In this study, we examine these two quality dimensions, where there is generally a trade-off between originality and utility (Rietzchel, Nijstad and Stroebe, 2019, as well as the surprise elicited by AUT responses as suggested by Boden (2004) and Simonton (2018). Surprise, where a response violates expectations and elicits interest, may be of particular interest when examining AI creativity in the context of LLMs. Given that LLMs are trained on nearly all the text on the Internet to -in essence- predict a text fragment given only the surrounding text, it would seem difficult for an LLM to generate a surprising, out-of-context response to the creative uses question.

A more recent method of gauging how creative a response is, is to measure the semantic distance of the response from the AUT object, a process automated with the SemDis software (Beaty and Johnson, 2021). SemDis measures are related to expert ratings of creativity and can be used as a proxy for creativity scoring (Beaty and Johnson, 2021).

Another method of analyzing a set of responses is focused more on the response process (Nijstad et al., 2010). Do most responses come from one conceptual space (e.g., using a brick as a paperweight, doorstop and bookend; so, to hold things in place)? Or is the response pattern more flexible, where numerous conceptual spaces are traversed (e.g., using a brick as a paperweight, sidewalk chalk and hot water bottle)? Hass (2017) found that whereas people often respond in clusters on semantic fluency tasks, such as listing as many animals as they can within a minute (e.g., naming pets, then naming farm animals, then zoo animals, etc.), people tend to use a more flexible strategy on the AUT. With GPT-3 being a predictive model, will it show a similar flexible response pattern?

To our knowledge this study represents the first systematic psychological assessment of a LLM's creativity. We compare how humans and GPT-3 score in terms of expert ratings of originality, usefulness and surprise of responses, automated semantic distance scoring as a proxy for creativity, and more holistically examine the flexibility of responses within a response set.

Methods

Sample

The human sample comprised of previously collected data of 823 responses from 42 students from the University of Amsterdam. Only data from students fluent in Dutch, the language of the study, were invited to participate. Written informed consent for participation was obtained and participants received study credits for participation. The data collection and subsequent re-use of the data was approved by our Psychology Department's Ethical Review Board (ERB number 6990).

The GPT-3 sample comprised of 144 runs of the AUT using the instructions and parameter settings described under Materials and Procedure.

Materials

Alternative Uses Task for humans

We used a computerized version of the Alternative Uses Test (AUT; Guilford, 1967). Participants were given the name of an object and instructed to "Think of as many creative uses for" the object as possible within a two minute period. In this study we use the data from the "Book", "Fork", and "Tin Can" objects. Participants were instructed to "Type each solution in the text box below and press Enter to add it to the list.". The solutions remained on the screen until the time limit was reached**.**

Alternative Uses Task for GPT-3

We used Open AI's API to request responses from GPT-3 for each of the same objects administered to humans: "Book", "Fork", and "Tin Can".

Before collecting GPT-3's AUT data for this study, we conducted two Monte Carlo experiments to determine: (1) which GPT-3 engine performed best on the AUT and (2) which prompt and parameter settings let to the most valid responses (i.e. responses that answered the question, did not contain nonsense) and provided the highest snapshot creativity scores (Silvia et. al., 2008). See osf.io/vmk3c/ for the code, data and results of our optimization studies.

Based on these results we administered the AUT to GPT-3's davinci-002 engine as follows. The instruction was: "What are some creative uses for a [book|fork|tin can]? The goal is to come up with creative ideas, which are ideas that strike people as clever, unusual, interesting, uncommon, humorous, innovative, or different. List [9|10] creative uses for a [book|fork|tin can]." The most important parameter settings that differed from the default were the temperature (sampled from range .65 - .80), the frequency penalty (set to 1), and the presence penalty (also set to 1). We collected 820 responses from GPT-3 over two sessions.

Procedure

Before we could score the responses we needed to make sure the judges could not easily distinguish between human and GPT-3 responses. First, we translated the 823 Dutch language human responses to English so that all responses were in the same language. Second, we removed characteristic punctuation from the GPT-3 data (e.g., numbered responses, period at the end of each line). Third, we systematically removed common phrases such as "Use a {object} to", "A {object} can be used to make", which is a step we usually take to make the rating of responses easier for human judges, which also happened to occur more often in the GPT-3 data.

After ensuring that GPT-3 and human responses were indistinguishable in all regards except main content, two trained judges rated each response on originality, utility, and surprise using a 5-point scale (from 1 = "not original | useful | surprising" to 5 = "highly original | useful | surprising") according to pre-specified scoring protocols. Judges were blinded to whether or not the responses stemmed from humans or GPT-3. The inter-rater agreement (assessed for approximately 10% of the 1656 responses) was ICC=.57 for originality, .68 for utility and .67 for surprise, which is considered fair to good. After removing invalid and incomplete responses (which were also excluded from analyses), the ICC's were .78 for originality, .70 for utility and .79 for surprise, which is considered good to excellent.

We computed the semantic distance (i.e., 1 – cosine similarity) between the vector embeddings for each response and the AUT object in question, which serves as a proxy for creativity (Beaty and Johnson, 2021), using the spaCy library and en_core_web_lg semantic space.

We also computed the flexibility of a response set. This was done by categorizing each response into one or more pre-defined categories and then dividing the number of categories in a response set by the number of responses in total in the response set. For example, if five responses were given to the AUT fork and each of these responses fell under the categories of "utensil" or "make music", then the resulting flexibility score would be 2 categories / 5 responses = 0.4 Three trained judges categorized the responses (one judge per object), by assigning each response to one or more pre-defined categories.

All data, code, Dutch to English translations, data cleaning steps, rating protocols, and categorization protocols are available on the Open Science Foundation website for this project: http://osf.io/vmk3c/.

Results

In total, after data cleaning, we analyzed 774 responses from 42 humans and 781 responses from 144 GPT-3 runs. All data and analysis code can be found on http://osf.io/vmk3c/.

1. Use a tin can as a mirror 2. to create toys 3. to create jewelry 4. as wallpaper as long as you stick enough next to each other 5. to throw 6. as a knife 7. as a key ring with the clip from the can 8. As a rattle with the clip in tin 9. as art 10. As a reminder of Andy Warhol	1. Plant a herb garden in tin cans 2. Make a wind chime out of tin cans and beads 3. as candle holders for an outdoor party 4. Create a mini Zen garden in a tin can 5. Make a robot out of recycled materials, including tin cans 6. Turn a tin can into a night light

Figure 1. Two sets of responses to the AUT "Tin Can" task. Can you guess which one was given by a human and which one by GPT-3?[1]

Do humans or GPT-3 provide more original, useful or surprising responses?

We used hierarchical regression models to predict originality, utility and surprise ratings and the semantic distance scores at the response level, while accounting for correlations between responses within a person or a GPT-3 run. The predictors were a human versus GPT-3 contrast plus the AUT object (3 levels).

As can be seen in Figures 2, 3 and 4, our results showed that humans had higher originality (β=.17, SE = .06, *z* = 2.91, *p* = .004) and surprise (β=.14, SE = .07, *z* = 1.96, *p* = .050) ratings as well as larger semantic distance scores (β= .10, SE = . 02, *z* = 5.45, *p*<.001) than GPT-3. Whereas GPT-3 had higher utility ratings (β=-.55, SE = .06, *z* = -8.52, *p*<.001), see Figure 5. In both groups, originality and utility were negatively correlated (r = -.56 for humans and r = -.61 for GPT-3).

Do humans or GPT-3 show more flexibility in their response patterns?

We computed flexibility scores for both humans and GPT-3 on the AUT tin can data. Humans had, on average, higher flexibility scores (*F*(1, 85) = 5.53, *p* = .021). However, as can be seen in Figure 6, GPT-3's flexibility scores show greater variance. GPT-3's flexibility scores were not related to temperature (*r* = .04, *p* = .80).

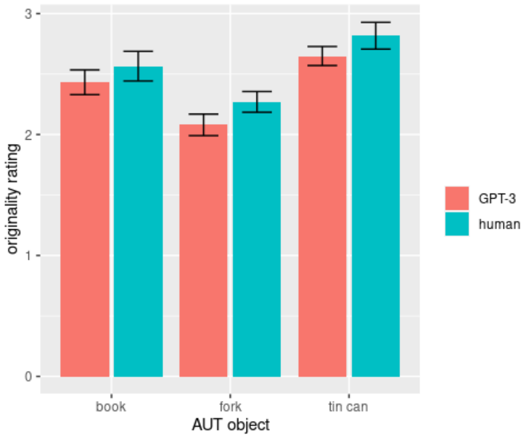


Figure 2. Human versus GPT-3 originality ratings. Human responses are rated to be more original.

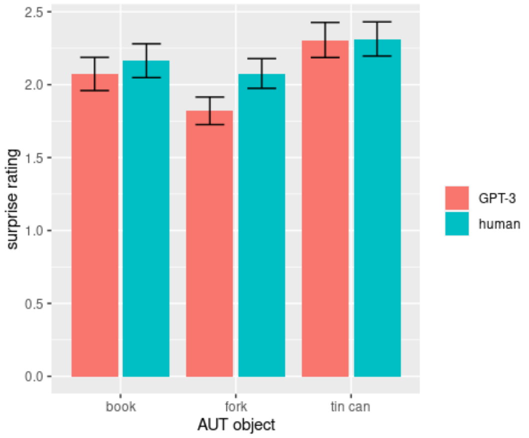


Figure 3. Human versus GPT-3 surprise ratings. Human responses are rated to be more surprising, but it's a close call.

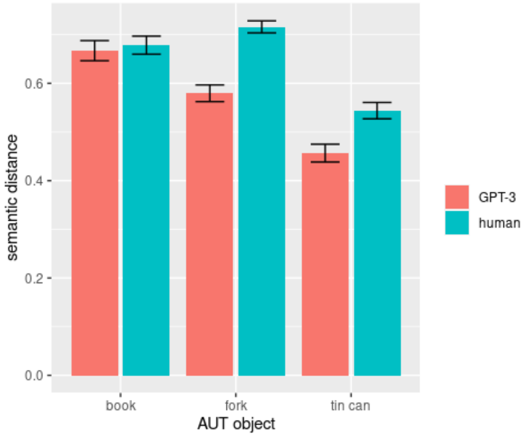


Figure 4. Human versus GPT-3 on semantic distance between response and AUT object embeddings, a proxy for creativity.

[1] In Figure 1, the human responses are on the left and GPT-3's responses are on the right.

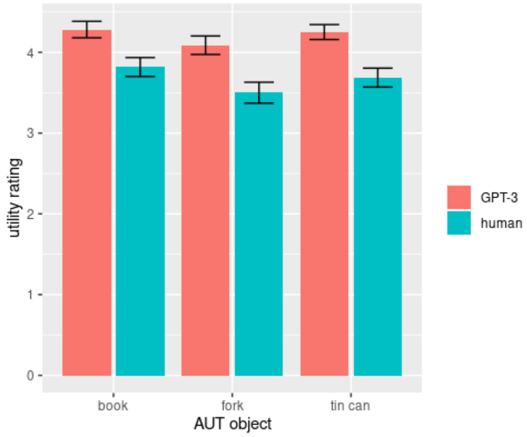


Figure 5. Human versus GPT-3 utility ratings. GPT-3 responses are rated to be more useful.

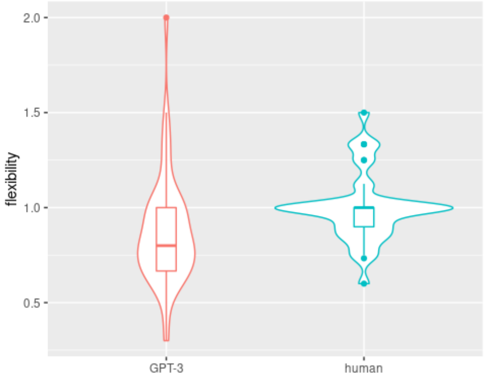


Figure 6. Human versus GPT-3 flexibility scores. Humans have a higher mean flexibility score, but GPT-3's scores show greater variance.

Discussion

This study aimed to put GPT-3 creativity to the test using the popular Alternative Uses Test (AUT, Guilford, 1967), where participants have to come up with creative uses for an everyday object. We compared a group of psychology students' performance to different runs with GPT-3 using the criteria creativity researchers generally assess on the AUT: the originality and usefulness of responses, as well as the often discounted component, surprise. Human responses were rated higher on originality and surprise. Similarly, the semantic distance scores between the AUT object and a response, which can be considered a proxy for creativity (Beaty and Johnson, 2021), were greater for human responses. However, GPT-3's responses were rated as more useful. In both groups, the originality-utility trade-off was apparent. In general, originality weighs in more when assessing creativity (Diedrich et al., 2015), so in this case the human responses would be considered more creative.

We also compared how flexible the response sets of humans and GPT-3 were, where flexibility was computed by dividing the number of categories present in the response set by the total number of responses. So, if a response set contained five responses stemming from three categories then the flexibility score was 3/5. Humans had higher flexibility scores, but there was greater variance in GPT-3 flexibility scores. It is unclear why GPT-3's flexibility scores are more variable; it is not a function of the temperature. We leave a more thorough investigation of the flexibility of responses for future work.

The main limitation of our study is the question of whether the Alternative Uses Task, a divergent thinking task, even measures creativity (Runco, 2008; Stevenson, Baas and van der Maas, 2020). Even assuming that it only measures one aspect of creativity, we believe that comparing AI and human performance can provide us with unique insights into what creativity is and how to best measure it.

Another limitation is that our Monte Carlo experiments to determine the best combination of instructions and parameters for GPT-3 to provide optimal creative responses were not as fine-grained as we would have liked. And, on the other hand, when we administered the Remote Associates Test (Mednick, 1968) and various creative insight problems (e.g., the egg problem, Sternberg and Davidson, 1982) we received responses that seemed to have been taken verbatim from journal articles or manuals. It appears likely that many creativity tests were present in GPT-3's training data.

A final limitation concerns our human sample, not only is it small, limited to college students, but it also consisted of native Dutch speakers. So, the results of this pilot study do not necessarily generalize to most humans. Also, in order to better compare GPT-3's responses to those of our human sample we had to translate the Dutch responses to English before scoring and analyzing the data. Some creative subtleties may have been lost in translation. Furthermore, humans received only minimal instructions, whereas we optimized instructions and parameters for GPT-3. Was it a fair fight? In future work we plan to administer the AUT and other newly developed creativity tasks with optimal instructions for both humans and AI and collect data in the same language.

At this point in time, we can conclude that GPT-3's performance on the AUT is not as creative as this sample of psychology students. But, at the same time GPT-3's performance is impressive and in many cases appears human-like. A Turing test is a logical next step in this line of research. We can imagine a future in which GPT-3 and other generative LLMs responses cannot be distinguished from humans, although the creative process will be different. This is where the question arises as to the role of process in defining what is creative and what is not; we agree with Boden (2004) and Simonton (2018), that the process matters, e.g., a brute-force process is not creative, but what is? We hope that this continued line of work will provide insight into what it means to be creative, and perhaps even what it means to be human.

References

Aalho, J., 2021. *Aum Golly – poems on humanity by an artificial intelligence.* https://aumgolly.fi/english/

Baer, J. and McKool, S.S., 2009. Assessing creativity using the consensual assessment technique. In *Handbook of research on assessment technologies, methods, and applications in higher education* (pp. 65-77). IGI Global.

Beaty, R.E. and Johnson, D.R., 2021. Automating creativity assessment with SemDis: An open platform for computing semantic distance. *Behavior research methods*, *53*(2):757-780.

Bender, E.M., Gebru, T., McMillan-Major, A. and Mitchell, S., 2021, March. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency.* 610-623. https://doi.org/10.1145/3442188.3445922

Boden, M. 2004. *The creative mind: Myths and mechanisms.* Routledge.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A. and Agarwal, S., 2020. Language models are few-shot learners. *Advances in neural information processing systems*, *33*, pp.1877-1901. https://arxiv.org/abs/2005.14165

Diedrich, J., Benedek, M., Jauk, E. and Neubauer, A.C., 2015. Are creative ideas novel and useful?. *Psychology of Aesthetics, Creativity, and the Arts*, *9*(1), p.35-40. https://doi.org/10.1037/a0038688

GPT-3. 2020. A robot wrote this entire article. Are you scared yet, human? *The Guardian.* https://www.theguardian.com/commentisfree/2020/sep/08/robot-wrote-this-article-gpt-3

Green, O., 2020. *Bob The Robot: Exploring the Universe - A Cozy Bedtime Story Produced by Artificial Intelligence*. Stolkholm: Olle Green.

Guilford, J.P., 1967. Creativity: Yesterday, today and tomorrow. *The Journal of Creative Behavior*, *1*(1):3-14.

Hass, R.W., 2017. Semantic search during divergent thinking. *Cognition*, *166*, pp.344-357. https://doi.org/10.1016/j.cognition.2017.05.039

Mednick, S.A., 1968. The remote associates test. *The Journal of Creative Behavior. 2*:213-214.

Mitchell, M., 2021. Abstraction and analogy-making in artificial intelligence. *Annals of the New York Academy of Sciences*, *1505*(1):79-101. https://doi.org/10.1111/nyas.14619

Nijstad, B.A., De Dreu, C.K., Rietzschel, E.F. and Baas, M., 2010. The dual pathway to creativity model: Creative ideation as a function of flexibility and persistence. *European review of social psychology*, *21*(1), pp.34-77. https://doi.org/10.1080/10463281003765323

Rietzschel, E. F., Nijstad, B. A., and Stroebe, W. 2019. Why great ideas are often overlooked. *The Oxford handbook of group creativity and innovation*, 179-197.

Runco, M.A., 2008. Commentary: Divergent thinking is not synonymous with creativity. *Psychology of Aesthetics, Creativity, and the Arts*, *2*(2):93–96. https://doi.org/10.1037/1931-3896.2.2.93

Runco, M.A. and Jaeger, G.J., 2012. The standard definition of creativity. *Creativity research journal*, *24*(1):92-96. https://doi.org/10.1080/10400419.2012.650092

Silvia, P.J., Martin, C. and Nusbaum, E.C., 2009. A snapshot of creativity: Evaluating a quick and simple method for assessing divergent thinking. *Thinking Skills and Creativity*, *4*(2):79-85. https://doi.org/10.1016/j.tsc.2009.06.005

Simonton, D.K., 2018. Defining creativity: Don't we also need to define what is not creative?. The Journal of Creative Behavior, 52(1), pp.80-90. https://doi.org/10.1002/jocb.137

Sternberg, R.J. and Davidson, J.E., 1982. The mind of the puzzler. *Psychology Today*, *16*(6):37-44.

Stevenson, C.E., Baas, M. and van der Maas, H., 2021. A minimal theory of creative ability. *Journal of Intelligence*, *9*(1):9. https://doi.org/10.3390/jintelligence9010009

van der Maas, H.L., Snoek, L. and Stevenson, C.E., 2021. How much intelligence is there in artificial intelligence? A 2020 update. *Intelligence*, *87*:101548. https://doi.org/10.1016/j.intell.2021.101548

Acknowledgements

We thank Emma van Lipzig for her contribution to the data categorization. This research was partly supported by the Jacobs Foundation Fellowship 2019-2022 awarded to Claire Stevenson (2018 1288 12).

Author Contributions

CS conceived and designed the study, collected the data, performed the analysis and wrote the original draft of the paper. IS and MB performed data coding and helped edit and review the camera ready paper. RG helped design the study and contributed to data analysis. HvdM helped conceive the study and provided supervision.

4. Generative art

Artistic Autonomy in AI Art

Alayt Issak and Lav R. Varshney
Coordinated Science Laboratory
Univerity of Illinois Urbana-Champaign
Urbana, IL 61801 USA
aissak@illinois.edu

Abstract

The concept of art has transposed meaning and medium across time, with its context being a deciding factor for its evolution. However, human beings' innermost functionality remains the same, and art, to this day, serves as an expression of the subconscious. Accelerated by the conception of GANs in 2014, automation has become a central medium in Artificial Intelligence (AI) Art. However, this raises concern over AI's influence on artistic autonomy within the process of creativity. This paper proposes ethical care towards maintaining the artist's volition in exercising autonomy in AI Art and utilizes principles of self-determination theory alongside fundamental limits of creativity to do so.

Introduction

Ethical care to creativity and intent

The traditional role of automation in society served to make human lives easier by outsourcing mundane tasks, and, very traditionally, to replace human jobs that would cut costs and increase profits. Recommender systems, for example, utilize language models to engage users in predictive text systems. However, much criticism has fallen on this medium as it alters the way people write. These systems have been found to make people "machine-like" – which is evident given its intention (Varshney 2020b). This prompts ethical care on the implementation of automation within attributes that characterize humanity—one of which is creativity.

Indeed as early as 1964, invoking Goethe's *Sorcerer's Apprentice*, the scholar of technics Lewis Mumford had argued: "let me first challenge the notion that automation is in any sense a final good, so beneficial in every aspect that the process must be hastened and extended relentlessly into every field . . . If the human organism had developed solely on that principle, . . . man would have been left without a thought in his head" (Mumford 1964).

In psychoanalysis, creativity serves as the expressive element or natural human impulse that drives the artistic experience (Zweig 2012). It is what drives surprise within viewers for pushing the boundary of what is deemed to be the experience of reality. It is also surprise that drives creativity as examined by its use for intrinsic motivation in creative action-taking as implemented by the artificial creative system of curious robots (Saunders et al. 2010). AI Art, with emphasis on its support to human creativity through creative machines, falls under criticism for automating this very process, given that the trade-off to maintain creative autonomy is evident in the practitioner.

Much work in Computational Creativity (CC) argues for the importance of process rather than just products of creativity (Colton 2008; Jordanous 2016), and further work, has introduced the humble creative as a means of furthering human development through co-creative processes that cultivate human creativity through its advanced creative capabilities (Cassion, Ackerman, and Jordanous 2021). This comes to show certain feats CC has taken in advancing co-creativity by alluding the working definition of CC towards responsibility that is detached from the artist.

As a result, this perspective goes in line with much CC work, where in creating tools that could in itself be deemed creative, has led to autonomous systems that extend beyond generative adversarial networks (GANs) such as The Painting Fool (Colton 2019). However, reciting back to process, we focus on the co-creative interaction of generative deep learning algorithms that are responsible in co-creation, and as such navigate the role of these algorithms with emphasis on Generative Adversarial Networks (GANs) due to their foundational blueprint to existent and advancing role in the contemporary AI artist's toolbox.

As an agent of play to enact creativity, GANs are utilized as a black box for providing artistic result, where the feedback loop is based on the artist's alteration of the algorithm upon interpretation of results. Other deep generative AI modeling techniques such as variational autoencoders and normalizing flows have also been used in the same manner. Unlike creation where artists decide meaning and form in process, this form of AI Art limits artistic autonomy by basing the artist's process upon output i.e. generating multiple sessions of training and determining the artwork based on generated artifacts. The limitations exhibited by this phenomenon has since led to interventions in the chain of computations, and is primarily exhibited by in-training modifications of intervening in the GAN latent space (Broad et al. 2021). We take these exceptions to recover human autonomy into account (as per our proposal for new ethics in AI Art), and present human-centric means that led certain practitioners to do so.

With regards to design intent, GANs were originally focused on improving quality, stability, and variation (Radford, Metz, and Chintala 2016) in order to implement the style transfer of the input image. Since then, they have evolved from representation to visually indeterminate artifacts to create an AI Art identity (Hertzmann 2020). However, the implementation of this medium still surrenders the creative process as the artifact's varied intent (Ventura 2016) does not address the fundamental loss in autonomy that occurs within automation (McCormack, Gifford, and Hutchings 2019). In June 2021, a discussion series on AI research and social responsibility, titled Post-Human Creativity: The Use of AI in Art, featured artists who emphasized the need to strengthen "interactions between humans and machines . . . instead of making technology more human" as to preserve "meaningful interactions with algorithms and push the boundaries of creative processes." (D<AI>DALOS 2021) With the concerns for AI's role in art in mind, we consider the ethical implications to the artist's creative autonomy via principles in self-determination theory and intent via fundamental limits of creativity.

Defining Creative Processes

Self-determination theory

Self-determination theory is a branch of psychology that suggests people are motivated to grow and change by three innate and universal psychological needs: autonomy, relatedness, and competence (Ryan and Deci 2000). Autonomy, or regulation by the self, is a phenomena that parallels other aspects of existence such as will, choice, and freedom. It is further augmented into liberty (independence from controlling principles) and agency (capacity for intentional action) (Ryan and Deci 2006).

We consider the limitation of AI Art to suffice liberty by considering abstraction in art as the mere generation of such artwork has led to a misuse of its abstract notion (Ventura 2016). In the style transfer of AI Art, artists often use forms that acquire a sense of talent, such as impressionism, to replicate the delicacy of the form's timeless novelty. However, when art is dictated in such sense, it transforms to a craft. Much like impressionism that emphasizes craftsmanship, AI Art then too becomes a craft that needs to be perfected through training, i.e. craftsmanship in training an AI model, which in the literal sense occurs via numerous iterations of training a model.

Historically, numerous iterations for craftsmanship was not the case. In 1979, Benoit Mandelbrot, a visionary mathematician and artist, introduced the "Mandelbrot set", a class of quadratic recurrence equations in the complex plane (Weisstein 2002). This development led to a renaissance of computer-generated art coined as *fractals*. Despite the recursive element that generates fractals, this early embodiment of computer-generated art was created to give form to mathematical revelation. The form was thus a byproduct of Mandelbrot's revelation of recursive structures revealing each other indefinitely, and can be attributed to his liberty to explore the depth of mathematics—a creative discipline much like art. Thus, as exemplified by early practitioners who embodied this liberty as a core element of their craft, current AI Art practitioners carry the responsibility of expanding their motive beyond sole mastery in order to embrace true creativity within the field.

On the other hand, taking a rather direct approach to abstraction in art, we explore creation that is rooted in Abstract Expressionism. Abstraction took time to develop appreciation due to the neglect for traditional talent per established artistic canons (Schwabsky 2009), let alone expressionism, which is expressive of the artist's inner feelings (Tejera 1965). In the 1950s, Abstract Expressionism led to two divergent stylistic tendencies: chromatic and gestural abstraction (National Gallery of Art 2022). In chromatic abstraction, the surrender to elements, such as color, shape and light, illuminate complexities to thought. For example, Mark Rothko painted what is simple, yet complex to express complexities in subtle form, see Figure 1.

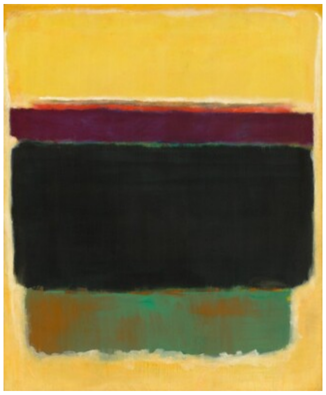


Figure 1: Untitled, Rothko

In his process, each abstraction held specific and original meaning, whereas modelling his form of creation via AI Art would not suffice as it would craft, but not hold meaning on the basis of liberty for the artist's expression, i.e. the artist's inner world. The expression would be decided upon the resultant AI abstraction, reversing art's role as revelation to form, as well as the practitioner's role from artist to audience.

In gestural abstraction, creativity spurs from the artist at the moment of creation and stems from the inner spark, or according to the psychoanalyst Carl Jung, "not accomplished by intellect but by play" or to a larger extent the "daimon of creativity" (Jung 1977). This moment, much like the deep immersion that comes with it, is encouraged and developed by a constant interaction that need not be interrupted, regulated, or automated (Diamond and May 1996). Hence, if one were to create AI Art based on gestural abstraction, such as Jackson Pollock's action painting (Solomon 2001), see Figure 2, then the artist would lose its creative autonomy because of artistic interruption during the surrender of process to AI.

Therefore, in both divergent cases of Abstract Expressionism, it is the human element of the artist that drives the possession of form, and as such frees the extremes and complexity of human consciousness (Grey and Wilber 2001). Whether subtle or spontaneous, for AI Art to emulate these

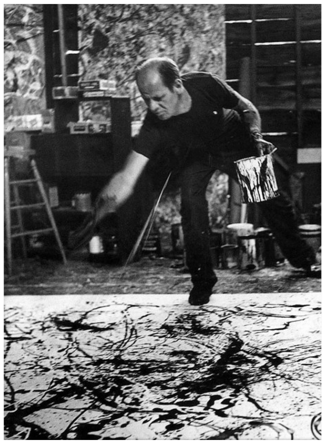


Figure 2: Action Painting, Pollock

works within its training corpus would lack its core essence in conveying the emotion of the artist and the resultant liberty needed for the process of creation.

Defining Design Intent

Fundamental limits of creativity

In one interpretation, intentionality is the inspiration or desire to express the human intent (Collingwood 2013). The capacity for this action is captured by the need for agency in autonomy. Fundamental mathematical limit theories for creativity have detailed a limit theorem whereby tradeoff between novelty and quality for a given creative domain exists (Varshney 2019). To consider a limit theorem for creativity with intentionality, Claude Shannon's capacity-cost-function formalism, which captures limits of reliable communication, is modified to address the semantic problem of creativity. Incorporating intentionality, semantic creativity shows that requiring communicative intent may reduce the quality and/or novelty of creative artifacts that are generated (Varshney 2020a).

In practice, this inverse relationship between intent and novelty is paralleled by examples in Dada art, such as Duchamp's fountain, that, despite the utmost intent, garnered controversy on the novelty of artistic creation (Hutchinson 2015). This begs to consider the role of novelty in AI Art due to the compromise of intent, in part of autonomy, as characterized by human creativity (McCormack, Gifford, and Hutchings 2019).

Indeed, it is accepted that human-level intentional creative autonomy for a system is difficult to achieve. With the failure of symbolic CC (to act from meaning), and embodied CC (through situated cognition), current practices allude to non-anthropocentric CC systems rooted in systems with intrinsic motivations of their own (Guckelsberger, Salge, and Colton 2017). In the minimal model presented to address this question, one argues a system must constitute autonomy and adaptivity (to exhibit a novel and valuable response to perturbation) in order to be necessarily creative. As this is yet to find its way in existing CC framework and literature, we allude to co-creative processes that fall in the current domain for what is fundamental to intent.

In theory, intent is highly discussed in Wassily Kandisky's book, *Concerning the Spiritual in Art*, via inner artistic elements. In his synopsis, the inner need of the artist is built up of three elements, namely every artist as a creator (element of personality); a child of the age (element of style), and a servant of art (element of pure artistry) (Kandinsky 1977). The second element of style details every artist to express the spirit of the age, alluding to the leverage of AI into art. However, this calls upon careful inquiry as borrowing of method by one art from another (AI Art from its predecessors), can only be truly successful when the application of the borrowed methods is not superficial but fundamental to the artist's endeavor (Spector 2018).

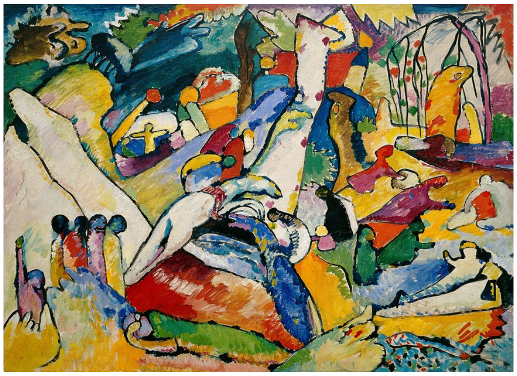


Figure 3: Sketch for "Composition II", Kandinsky

For example, adapting of form to its inner meaning in Kandinsky's Sketch for "Composition II" which rids conventional aesthetic values for his time, as seen in Figure 3 above, cannot be the basis of visual indeterminacy of AI Art as it must find its own form to its inner meaning. Thus, in order to move beyond novelty, AI Art must incorporate the artist's inner and essential elements as producer (Jordanous 2016) to harness AI as a creative medium and create what is fundamental to its age.

New Ethics to Autonomy

We now propose a new ethics for artistic autonomy in AI Art that focuses on co-creative processes in line with our human-centric approach to autonomy. Accordingly, we present concrete ways to re-center human creativity and intentionality when co-creating with AI systems by attending to the approach of Collaborative AI, i.e. systems designed to support the creative practice of human artists (D'Inverno and McCormack 2015).

Re-centering creativity

To re-center creativity between AI and the human artist that will create fundamental art, the artist needs interaction, feedback, reminding, connection, stimulation and interaction from its AI partner (D'Inverno and McCormack 2015). An important tool that has opened doors and accelerated this connection has been multimodal prompt programming, or programming in natural language for text-to-image synthesis, which originated in January 2021 with the release of the novel CLIP+VQGAN framework (Miranda 2021).

Not only did this framework democratize and increase accessibility to AI Art, but it also opened a new paradigm for natural language interaction with AI, much like the conversational interactions one would have with a human being that is deemed to be intelligent. The personification of the tool with natural language interaction has allowed AI artists to develop their own creative practice through a humanistic interaction via prompts that probe the generative model (VQGAN). As a result, this interaction has challenged, provoked and supported artists with re-centered creativity to synthesize images in way they are stimulated to do so.

To elicit re-centered creativity in prompt programming furthermore, we highlight distinctions offered by two of the four perspectives on computational creativity (Jordanous 2016). In process, the artist can take a holistic approach to image generation by viewing the synthesis of images at each time step as part of the creative process. Thus, re-centered creativity emerges for which the artist may even choose the desired image based on the emotion it invokes regardless of the training iteration. For the aforementioned exceptions, this is paralleled by intervening in the GAN latent space. Whereas in product, one can direct synthesized images to a desired environment that it deems beneficiary. For instance, a recent children's book set to expand upon a child's imagination with the expansive abstractions generated using prompt programming techniques (Issak and Varshney 2022).

Re-centering intent

To re-center intent in AI Art, one consideration would be to rethink novelty and aim for a simultaneous increase of creative autonomy and intent by alluding to a co-creative process that hands over creative autonomy. Although some argue this can only be possible given Metacreativity (giving creative autonomy to CC systems that possesses a self), the trade-off here alludes to aspects where automation in AI art is necessary for creative autonomy, and thus implements it to fulfill what one may not possess (Berns et al. 2021).

For instance, DARCI (Digitial ARtist Communicating Intention) is a creative system that exhibits creativity by noting the attribution of creativity with respect to system intentionality and autonomy (Ventura 2019). It has, thus far, addressed these difficulties and maintained to exhibit these characteristics to some extent (Norton, Heath, and Ventura 2010). Drawing back to the "black box" analogy for AI training and the resultant novelty, one may then consider the integration of intent within the co-creative process system by assigning the loss in novelty towards the artist.

In one way, this consideration can reveal surprising results that automation can afford. For example, the art collective aurèce vettier reinvents intent by exploring hybrid combinations of art and algorithms. In their work titled, *Brightly-Lit Stool, Four-eyed Cat*, see Figure 4, the collective displays a "painting/technology" to expand conceptual possibilities of AI Art. In doing so, they curate a dataset beginning from personal photos of their pet cat, generate images which wound up distorted in part of the training process, intentionally pick one in which a four eyed cat emerges, and transform the chosen image onto a canvas for painting (aurèce vettier 2020). This way, AI serves as a tool to create a component of the entire piece, whereas novelty arises out of the artist's greater autonomy to create meaningful interactions with algorithms that push the boundaries of creative processes.

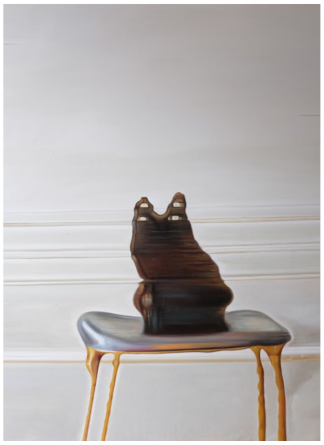


Figure 4: Brightly-Lit Stool Four-eyed Cat, aurèce vettier

Conclusion

The novelty of AI Art need not arise out of appreciation for AI's capability to create such works, but rather ask what the artwork entails in creativity and evidences in intent. As such, we encourage artists to re-calibrate the role of AI in their art by retaining their personal vision with an authentic foundation of creativity and intent (Grey and Wilber 2001). As proposed, such foundations may reveal the nature of the artistic process, incorporate room for interaction, explore insightful curiosity and perhaps unlock an inner creative in part of retrieved autonomy within the process of creation.

Future Work

While establishing ethics for re-centering creativity and intent, we also present the question of gestural abstraction in AI Art as it is yet to be addressed in the CC community. In line with our argument, perhaps this could be answered by rethinking the co-creative process for this art form. As this could be revealed in existing or future CC literature, we will keep these discussions in our thoughts.

Author Contributions

All authors participated in the curation, writing and fruition of this manuscript equally.

Acknowledgments

Discussions with theorists and practitioners at the 2021 Machine Learning for Creativity and Design workshop, NeurIPS, as well as reviews from IJCAI and ICCC are greatly appreciated.

References

aurèce vettier. 2020. Brightly-Lit Stool, Four-eyed Cat, AV-2020-U-70.

Berns, S.; Broad, T.; Guckelsberger, C.; and Colton, S. 2021. Automating Generative Deep Learning for Artistic Purposes: Challenges and Opportunities. *CoRR* abs/2107.01858:10.

Broad, T.; Berns, S.; Colton, S.; and Grierson, M. 2021. Active Divergence with Generative Deep Learning – A Survey and Taxonomy. Technical Report arXiv:2107.05599, arXiv. arXiv:2107.05599 [cs] type: article.

Cassion, C.; Ackerman, M.; and Jordanous, A. 2021. The humble creative machine. In *Twelfth International Conference on Computational Creativity, ICCC?21*. Association of Computational Creativity.

Collingwood, R. G. 2013. *The Principles of Art*. Read Books.

Colton, S. 2008. Creativity versus the perception of creativity in computational systems. In *Proc. AAAI Spring Symp., Creative Intell. Syst.*, 14–20.

Colton, S. 2019. From Computational Creativity to Creative AI and Back Again.

D<AI>DALOS. 2021. D<AI>DALOS – Post-Human Creativity: The Use of AI in Art | Artificial Intelligence Center | CTU Prague.

Diamond, S. A., and May, R. 1996. *Anger, Madness, and the Daimonic: The Psychological Genesis of Violence, Evil and Creativity*. State University of New York Press, 1st edition edition.

D'Inverno, M., and McCormack, J. 2015. Heroic versus collaborative ai for the arts. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, 2438–2444. AAAI Press.

Grey, A., and Wilber, K. 2001. *The Mission of Art*. Boston, Mass.: Shambhala, 1st edition edition.

Guckelsberger, C.; Salge, C.; and Colton, S. 2017. Addressing the "Why?" in Computational Creativity: A Non-Anthropocentric, Minimal Model of Intentional Creative Agency. In *Proceedings of the 8th International Conference on Computational Creativity (ICCC'17)*, 125 – 136.

Hertzmann, A. 2020. Visual Indeterminacy in GAN Art. *Leonardo* 53(4):424–428. arXiv: 1910.04639.

Hutchinson, M. 2015. Dada Contra Art History. *Dada/Surrealism* 20:1–22.

Issak, A., and Varshney, L. 2022. *Young McDonald Had a Botanical Farm*. Kindle Direct Publishing.

Jordanous, A. 2016. Four PPPPerspectives on computational creativity in theory and in practice. *Connection Sci.* 28(2):194–216.

Jung, C. G. 1977. *Symbols of Transformation*. Princeton, NJ: Princeton University Press, 2nd edition.

Kandinsky, W. 1977. *Concerning the Spiritual in Art*. New York: Dover Publications, revised edition edition.

McCormack, J.; Gifford, T.; and Hutchings, P. 2019. Autonomy, Authenticity, Authorship and Intention in computer generated art. *EvoMUSART 2019: 8th International Conference on Computational Intelligence in Music, Sound, Art and Design*.

Miranda, L. J. 2021. The illustrated vqgan. *ljvmiranda921.github.io*.

Mumford, L. 1964. The automation of knowledge. *AV Commun. Rev.* 12(3):261–276.

National Gallery of Art. 2022. Mark Rothko: Introduction.

Norton, D.; Heath, D.; and Ventura, D. 2010. Establishing Appreciation in a Creative System. In *Proceedings of the International Conference on Computational Creativity*, 10.

Radford, A.; Metz, L.; and Chintala, S. 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv:1511.06434 [cs.LG].

Ryan, R. M., and Deci, E. L. 2000. Self-Determination Theory and the Facilitation of Intrinsic Motivation, Social Development, and Well-Being. *American Psychologist* 11.

Ryan, R. M., and Deci, E. L. 2006. Self-regulation and the problem of human autonomy: does psychology need choice, self-determination, and will? *Journal of Personality* 74(6):1557–1585.

Saunders, R.; Gemeinboeck, P.; Lombard, A.; Bourke, D.; and Kocaballi, A. B. 2010. Curious whispers: An embodied artificial creative system. In *Proceedings of the International Conference on Computational Creativity, ICCC-10*.

Schwabsky, B. 2009. The Resistance of Painting: On Abstraction. *The Nation*.

Solomon, D. 2001. *Jackson Pollock: A Biography*. New York : Lanham, Md: Cooper Square Press, 1st cooper square press ed edition edition.

Spector, N. 2018. Sketch for "composition ii" (skizze für "komposition ii").

Tejera, V. 1965. *Art and human intelligence*. New York,: Appleton-Century-Crofts. Pages: xii, 237 pages.

Varshney, L. R. 2019. Mathematical Limit Theorems for Computational Creativity. *IBM Journal of Research and Development* 63(1):2:1–2:12.

Varshney, L. R. 2020a. Limits Theorems for Creativity with Intentionality. *In Proceedings of the Eleventh International Conference on Computational Creativity (ICCC)* 390–393.

Varshney, L. R. 2020b. Respect for human autonomy in recommender systems. arXiv:2009.02603 [cs.CY].

Ventura, D. 2016. Mere generation: Essential barometer or dated concept? In Pachet, F.; Cardoso, A.; Corruble, V.; and Ghedini, F., eds., *Proceedings of the Seventh International Conference on Computational Creativity, ICCC 2016, UPMC, Paris, France, June 27 - July 1, 2016*, 17–24. Sony CSL Paris, France.

Ventura, D. 2019. Autonomous Intentionality in Computationally Creative Systems. In *Computational Creativity*. Springer, Cham. 49–69.

Weisstein, E. W. 2002. Mandelbrot Set. Publisher: Wolfram Research, Inc.

Zweig, S. 2012. *The Struggle with the Daemon: Hölderlin, Kleist and Nietzsche*. Pushkin Press.

The @artbhot Text-To-Image Twitter Bot

Amy Smith[1] and **Simon Colton**[1,2]

[1] School of Electronic Engineering and Computer Science, Queen Mary University London, UK
[2] SensiLab, Faculty of Information Technology, Monash University, Australia

amy.smith@qmul.ac.uk s.colton@qmul.ac.uk

Abstract

@artbhot is a Twitter bot that brings the generative capabilities of CLIP-guided GAN image generation to the public domain by transforming user-given text prompts into novel artistic imagery. Until recently, access to such image synthesis techniques has been largely restricted to Google Colab notebooks, which require some technical knowledge to use, and limited services which require access. @artbhot increases access to text-to-image technology, as Twitter users already have the platform knowledge needed to interact with the model. We discuss here some of the technical challenges of implementing such a system, and provide some illustrative examples of its usage. We further discuss what this mounting of generative technology amongst social media could mean for autonomous computationally creative agents.

Introduction

Recent developments with generative deep learning technologies have enabled text-to-image computational models to produce artistic images and video content, given only text prompts from users. Colton et. al (2021) explored the possibilities for this, within the context of *generative search engines*, where images are generated rather than retrieved as per Google image search. Such approaches in the field of text-to-image synthesis (Agnese et al. 2019), allow the user to encode text in such a way as to drive a search for a latent vector input to a pre-trained image generation neural model. This technology has an impressive ability to innovate novel visual content from text, producing high quality and diverse imagery which reflects the prompt well, with images that are often surprisingly innovative. Examples of the kind of artwork that can be produced are given in (Smith and Colton 2021), and we describe the CLIP-Guided VQGAN text-to-image system in the background section below.

Interaction with such systems has been largely limited to Google Colab notebooks (Bisong 2019), but this has barriers to entry due to the the technical knowledge required to run the notebooks, and user interaction is limited to an image retrieval service. Other recent text-to-image generators (mentioned below) have invitation-only limited access for a small number of artists and researchers. To address this lack of access, we have built the @artbhot twitter-bot (Veale and Cook 2018), which embeds CLIP-guided VQGAN in the Twitter social media platform experience. As described and illustrated with examples below, people can tweet their text prompt with appropriate annotations, and expect an image to be returned in due course. This greatly increases accessibility to the public, as Twitter has over 200 million active users. Due to it's popularity and reach, and both the data and interaction available through its API, Twitter also provides an ideal platform for @artbhot to take on more creative autonomy. In particular, we plan to challenge the assumption that text-to-image users should be served only imagery which purely reflects their prompt. Instead, as described in the final section below, we aim for @artbhot to use prompts as springboards for creative ideation and visualisation and for it to enter into a dialogue with users in a fashion akin to discussions with artists on social media.

Background

In early 2021, Ryan Murdock combined OpenAI's Contrastive Learning Image Pretraining model (CLIP) (Radford et al. 2021) with the BigGAN generative adversarial network (Brock, Donahue, and Simonyan 2019) into a text-to-image generation process. He made the system available via a Colab notebook called *The Big Sleep*. In overview (with further details in (Colton et al. 2021)), the process involves first encoding a user-given text prompt into the CLIP latent space as vector v_1. Then the system performs a search for a latent vector input to BigGAN, v_2, which produces an image that, when encoded into the CLIP latent space as v_3, has optimally low cosine distance between v_1 and v_3. The search is performed using gradient descent to minimise a loss function based on this cosine distance. Given that related images and text are encoded by CLIP to similar places in the latent space, this approach tends to produce images which somehow reflect the given text prompt.

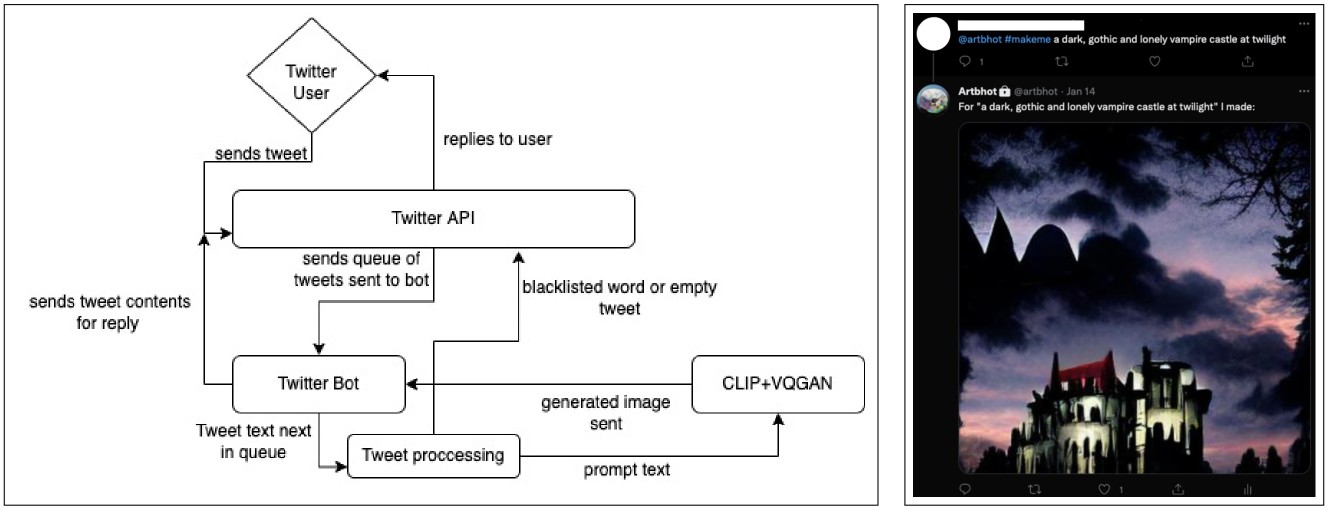
In the interim, many CLIP-guided text-to-image generators have been made available, with steadily improved quality and fidelity (with respect to the prompt) of the images produced. The most recent, and impressive examples of this generative technology are *@midjourney*[1], *Disco Diffusion*[2], DALL-E [3] from OpenAI and Imagen[4] from Google. DALL-

[1] midjourney.co

[2] tinyurl.com/yckn4h7

[3] openai.com/dall-e-

[4] imagen.research.google/



[redacted]
@artbhot #makeme a dark, gothic and lonely vampire castle at twilight

Artbhot @artbhot · Jan 14
For "a dark, gothic and lonely vampire castle at twilight" I made:

Figure 1: (a) Processing of a tweet by @artbhot (b) Example user interaction on Twitter.

E is particularly impressive as it employs a one-shot process, with an encoded text prompt fed-forward through a model to produce images near-instantaneously. However, the trained model is so large that access is limited, with the expectation that OpenAI will provide a subscription service for it soon. Currently, *Disco diffusion* is available as a Google Colab notebook, and @*midjourney* is only available to selected users. *Wombo Dream*[5] however is an app that is available for free from the app store, and appears to have been very popular. In addition to users being able to enter a prompt and receive an image based on this text, they can also select from several art styles that can influence the aesthetic of their generated image. These styles include 'Dark Fantasy', 'Mystical' and 'Salvador Dali'. There is also now *DALL.E mini* [6] which is available to the public and free of charge. It is a smaller version of the model mentioned above and is hosted on Hugging Face[7].

In a similar process to that of the Big Sleep approach, CLIP-guided VQGAN harnesses the perceptual power of CLIP and the image generation capabilities of the Vector Quantized Generative Adversarial Network (VQGAN) (Esser, Rombach, and Ommer 2021). This GAN architecture combines two approaches to interpreting meaning, using both discrete and continuous representations of content (Cartuyvels, Spinks, and Moens 2021). Discrete representations model a more human way of interpreting meaning aside from a pixel based approach, which is traditionally how computers have processed images. In particular it considers the image as a whole and interprets the relationships between the different compositional elements of the contents, i.e., relationships between different parts of an image (such as the sky and the ground in a landscape image).

VQGAN models these discrete representations as long range dependencies, meaning it can interpret the *relationships* between compositional elements, and not just the elements themselves, as described in (Esser, Rombach, and Ommer 2021). VQGAN models image elements, and the local relationships within visual parts of an image, using continuous representations (such as the RGB channels in a pixel). It also interprets discrete representations within image content using a transformer (Vaswani et al. 2017), but before a feature map can be passed to this, the model learns an intermediary representation of this image data using a *codebook*, as described at tinyurl.com/2vm3t9r8. This is a fixed size table of embedding vectors that is learned by the model. This intermediary stage is necessary, as transformers scale the length of an input sequence quadratically, making even a 224 x 224 pixel image above the processing capacity of most GPUs. CLIP-guided VQGAN is described in (Crowson et al. 2022), and various notebook for CLIP-guided VQGAN have been implemented, with a list of ten given here: ljvmiranda921.github.io/notebook/2021/08/11/vqgan-list/

@artbhot Implementation and Deployment

Twitter bots are usually small, autonomous programs running on a server, which regularly produce and tweet outputs composed of texts, images, animations and/or music/audio compositions, as described in (Veale and Cook 2018). More advanced bots can respond to replies on Twitter and/or tweets if they are hashtagged appropriately. Our Twitter bot, @artbhot, is currently only reactive, in that it is used as a service: people tweet text prompt requests at it, and it responds with a reply comprising an image that (hopefully) reflects the prompt, and a repetition of the prompt.

@artbhot is comprised of two parts: the generative process, which is provided by CLIP-guided VQGAN; and code which enables it to interact with the Twitter API. The implementation is hosted on a remote server which runs 24 hours a day, so users can access image generation capabilities on demand. Users can read instructions on how to use the bot from a document linked in the bio section of the @artbhot's

[5] wombo.art

[6] tinyurl.com/4eyr5yjv

[7] huggingface.co

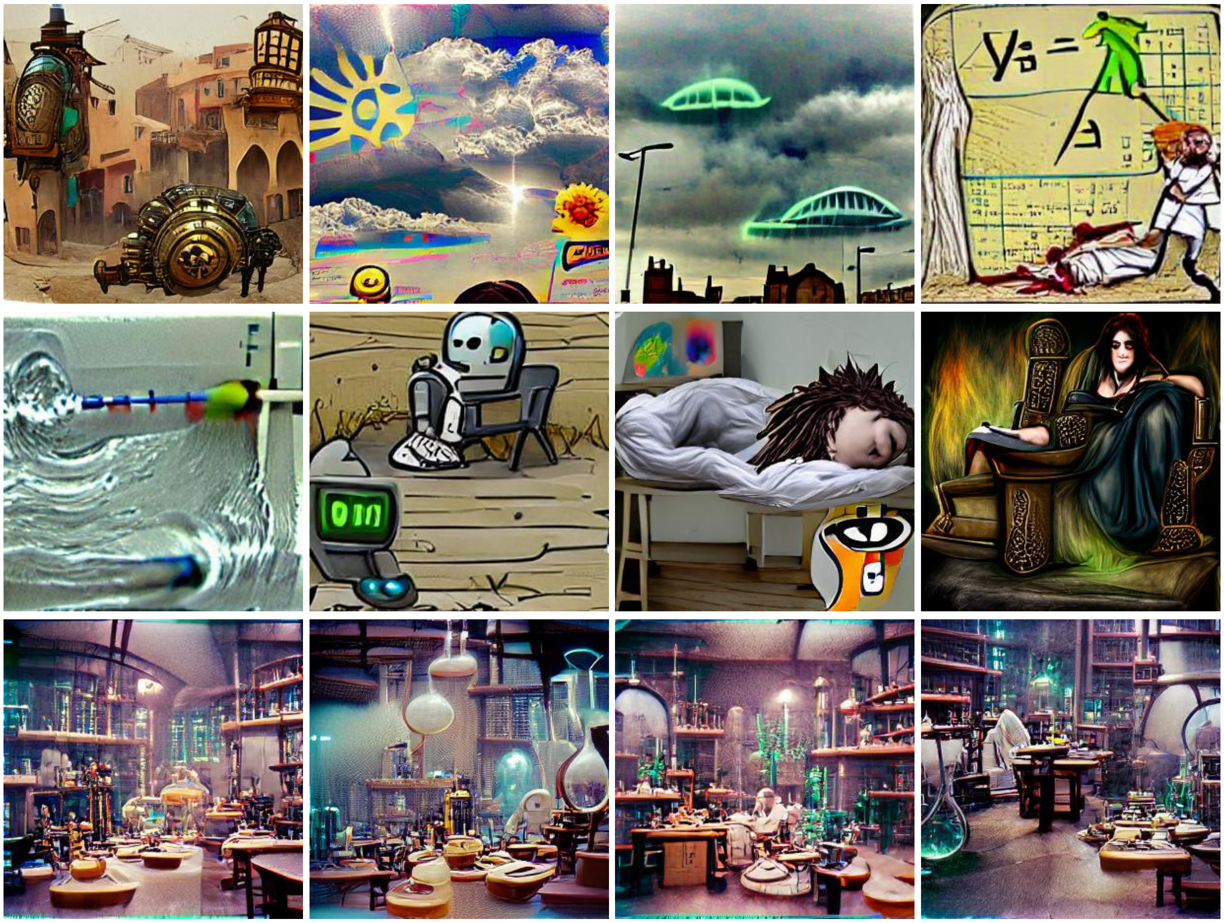


Figure 2: Generated images for prompts. **Top row:** “Steampunk morocco, concept art”; “🌞☀️🌤️🌦️🔆⛅”; “Aliens invading Newcastle Upon Tyne”; “Pythagoras killing his student because the square root of 2 is irrational”. **Middle row:** “A positive lateral flow test”; “Waiting for the bot”; “Wake up @artbhot”; “The Scribe, sitting in her throne. Deviant art character illustration”. **Bottom row (all):** “A 35mm analog film photo of an alchemists lab in the distant future”.

Twitter page. These instructions include how to communicate with the bot using the following tweet format:

```
@artbhot #makeme prompt text
```

(e.g. @artbhot #makeme an oil painting of a burger).

Every 15 seconds, the bot code checks for new tweets in this format from any user, using the python Twitter API. Once found, the prompt text is extracted, processed and either used as input for a CLIP-guided VQGAN process, or rejected for containing any prohibited words. This cross-referencing of the prompt against a list of prohibited words aims to keep the experience of using the bot as friendly as possible. If a prohibited word is found, a textual reply is automatically generated and sent to the user as a reply to their tweet, asking them to try again. The processing performed by @artbhot for a given input tweet is portrayed in fig. 1(a).

If an image is generated, it is then sent to the user via the Twitter API as a reply to their initial tweet, with a reminder of the prompt they used (this is to ensure that the prompt text follows the generated image in the case where a bot reply is shared on Twitter without the original tweet from the user to provide context). An example user interaction on Twitter with @artbhot is given in figure 1(b). The first iteration of @artbhot incorporated CLIP guided BigGAN for image generation, as this model was one of the best CLIP guided GANs available to the public. This was a local version of the code released in the Big Sleep colab notebook, installed on our server. Later, an implementation of CLIP-guided VQGAN was released (github.com/nerdyrodent/VQGAN-CLIP). On experimenting with this text-to-image generator, we found that the output from the newer model showed improvements in multiple ways. Firstly, almost no images were outright failures from VQGAN in the way that BigGAN regularly generated blank or highly noisy/textured uninterpretable images. Also, the fidelity of the image to the prompt was usually much better and there was much less visual indeterminancy (Hertzmann 2020), making the images more coherent from VQGAN than from BigGAN. For these reasons, we replaced BigGAN in @artbhot with VQGAN. The top two rows of figure 2 show 8 example images generated in response to tweets sent to it, which we refer to in the next subsection.

A Preliminary Evaluation

We plan to make @artbhot open to the public in 2022, after some additional implementation described in future work below. Before this, we have made it available to a user group of 16 people. It has been running for 5 months and has processed over 600 tweets, taking, on average, around 2 minutes for a user to receive an image in response to their tweet. While there have been no outright failures where images don't reflect the prompt at all, after an informal evaluation (by ourselves) of the most recent 100 replies to Twitter prompts, we found 16% of the images were not visually coherent enough to reflect the prompt satisfactorily. Two examples of this can be seen on the left of row two in figure 2, with neither properly reflecting the prompt "A positive lateral flow test" or "Waiting for the bot". Generally, the images that are less successful have a high degree of visual indeterminacy (Hertzmann 2020), making it difficult to interpret the content of the image and how it may be associated with the tweet text. Other factors for relative failure include content that is off topic, inaccurate colours for the subject matter, or image content that is too small and/or off-centre. We do acknowledge however that this is a subjective evaluation and that other opinions may differ regarding interpretations of image content.

We found that @artbhot was able to handle unexpected prompts, for instance ones containing emojis. As per the second image in the first row of figure 2, CLIP-guided VQGAN interpreted the weather emojis correctly and produced an image with sun and clouds. Diversity was also a concern, as users would expect a variety of images for similar prompts. We asked four users to each use the prompt "a 35mm analog film photo of an alchemists lab in the distant future", with the resulting images portrayed in the bottom row of figure 2. We see that there is some diversity, but perhaps not enough to be satisfying, and this is something we hope to improve upon, probably with automated augmentation/alteration of prompts.

Overall, the interactions users have had with @artbhot have been playful and casual, with people feeling free to try out all manner of interesting and unusual prompts, often trying to stretch the bot past its limitations. The qualitative responses we've gathered have been largely positive, with people reporting they have used it for amusement, entertainment and conversation, but wish it would return images faster, as attention can wane. We noticed some trends in the kinds of prompts users sent, including: referring to the bot itself (see middle row of figure 2); setting moods or styles such as *steampunk* (first image of top row); setting up imaginary or historical scenes such as aliens over cityscapes or pythagorean murders (top row, right); and asking for design inspiration (final image on the middle row). One user wanted longer interactions with @artbhot, in particular to ask it to enhance images and to combine their prompts/images with those from friends.

Conclusions and Future Work

Text-to-image colab notebooks are very popular, and initial responses to @artbhot suggest that it would also be very popular on twitter. Unfortunately, it is beyond our computational resources to provide GPU processing to anyone on twitter who tweets a prompt. Moreover, as predicted in (Colton et al. 2021), there seems little doubt that consumer text-to-image generation services will become available soon, and will likely find their way into products such as Adobe's Creative Suite eventually. For these reasons, we are interested in offering more than a service which fulfils image generation requests, as @artbhot currently does. Instead, we will open up @artbhot so that it can receive tweets from any member of the public (which it currently does not), and select a few tweets each day to reply to that have the highest potential for a meaningful, creative and thought-provoking interaction with the user. Once a user is selected, this longer interaction with @artbhot may take the form of a string of iterations on an image; as the user asks to 'evolvethis' image to repeatedly evolve the image with new prompts. This may also take the form of merging several tweets in to a prompt, that is then used to generate an image, using a 'mergethis' hashtag. In this way, the user will still feel in control of the process, but will receive innovative and surprising output as the bot takes on more autonomy.

On responding to the chosen prompts, we plan for @artbhot to apply a range of generative techniques and appeal to a number of computational creativity theories and practices. These include (on the text side) fictional ideation, humour, narrative generation, poetry, etc., and (on the imagery side) style transfer, animations, and visual stories. @artbhot will employ framing and explainable computational creativity techniques (Llano et al. 2020) to get users to look more closely at its ideas and creations. We further aim to enable @artbhot to learn from feedback, so as to be more interesting and engaging for users.

We also aim to encourage conversation and collaboration with users, to ultimately generate pieces deemed to be artworks rather than just imagery reflecting text. To do this, we will need to utilise existing evaluation techniques from casual creators (Compton and Mateas 2015) and computational creativity in general, and to develop new ones specific to the project. We will also need to implement more advanced artistic image generation techniques. We have already taken first steps in this direction by writing software which takes animations from @artbhot and makes a large collaged animation (as per fig. 3) for an exhibition[8] at the Pablo Gargallo Museum in Zaragoza, Spain; celebrating the life and work of nobel laureate Santiago Ramon y Cajal.

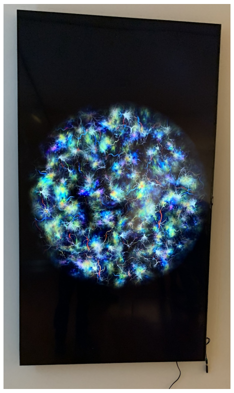


Figure 3: Exhibition piece: Pericellular Nests

[8] zaragoza.es/sede/servicio/cultura/evento/232731

Author Contributions

AS is lead author, SC is second author and contributed to writing, supervision, reviewing & editing. Both AS and SC contributed to the concept of @Artbhot, AS developed and evaluated @Artbhot. SC implemented initial interactions with CLIP + BigGAN while AS implemented initial interactions with CLIP + VQGAN.

Acknowledgments

We would like to thank Dr. Mike Cook for his support. We would also like to thank Ryan Murdock, Katherine Crowson and Nerdy Rodent for their ground breaking contributions in pioneering CLIP guided image generation. We would like to thank the anonymous reviewers for their helpful comments, and also thank the participants of the pilot study from the GSIG Discord group.

References

Agnese, J.; Herrera, J.; Tao, H.; and Zhu, X. 2019. A Survey and Taxonomy of Adversarial Neural Networks for Text-to-Image Synthesis. arXiv: 1910.09399.

Bisong, E. 2019. Google colaborator. In *Building Machine Learning & Deep Learning Models on Google Cloud Platform*. Springer.

Brock, A.; Donahue, J.; and Simonyan, K. 2019. Large Scale GAN Training for High Fidelity Natural Image Synthesis. arXiv: 1809.11096.

Cartuyvels, R.; Spinks, G.; and Moens, M.-F. 2021. Discrete and continuous representations and processing in deep learning: Looking forward. *AI Open* 2:143–159.

Colton, S.; Smith, A.; Berns, S.; Murdock, R.; and Cook, M. 2021. Generative search engines: Initial experiments. In *Proceedings of the International Conference on Computational Creativity*.

Compton, K., and Mateas, M. 2015. Casual creators. In *Proceedings of the International Conference on Computational Creativity*.

Crowson, K.; Biderman, S.; Kornis, D.; Stander, D.; Hallahan, E.; Castricato, L.; and Raff, E. 2022. Vqgan-clip: Open domain image generation and editing with natural language guidance. *arXiv:2204.08583*.

Esser, P.; Rombach, R.; and Ommer, B. 2021. Taming Transformers for High-Resolution Image Synthesis. arXiv: 2012.09841.

Hertzmann, A. 2020. Visual Indeterminacy in GAN Art. *Leonardo* 53(4):424–428.

Llano, M. T.; dÄôInverno, M.; Yee-King, M.; McCormack, J.; Ilsar, A.; Pease, A.; and Colton, S. 2020. Explainable Computational Creativity. In *Proceedings of the International Conference on Computational Creativity*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. arXiv: 2103.00020.

Smith, A., and Colton, S. 2021. CLIP-Guided GAN Image Generation: An Artistic Exploration. In *Proceedings of the EvoMusArt conference*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, .; and Polosukhin, I. 2017. Attention is All you Need. In *Advances in NeurIPS*,

Veale, T., and Cook, M. 2018. *Twitterbots: Making Machines that Make Meaning*. MIT Press.

Toward Modeling Creative Processes for Algorithmic Painting

Aaron Hertzmann
Adobe Research
hertzman@dgp.toronto.edu

Abstract

This paper proposes a framework for computational modeling of artistic painting algorithms, inspired by human creative practices. Based on examples from expert artists and from the author's own experience, the paper argues that creative processes often involve two important components: vague, high-level goals (e.g., "make a good painting"), and exploratory processes for discovering new ideas. This paper then sketches out possible computational mechanisms for imitating those elements of the painting process, including underspecified loss functions and iterative painting procedures with explicit task decompositions.

Introduction

In this paper, I describe aspects of human creativity and creative practice missing from current computational formulations, and sketch possible ways these ideas could be incorporated into algorithms. Perhaps by basing algorithms on human creative processes, we could develop new kinds of tools for artists. Such algorithms could even shed light on the mechanisms of human creativity.

This paper begins with several examples from expert artists' processes across different art forms, together with my own experience. These examples illustrate two main points. First, creative processes are often driven by **vague, high-level goals**, such as "make a good painting." Existing formulations treat an artwork as deriving from predetermined styles and goals. This paper argues the opposite: often, an artwork's apparent goals and style emerge from the creative process. Second, **exploratory processes** play a key role: different painting strategies will lead to different outcomes, rather than being purely a function of goals. Indeed, artists frequently discuss the importance of process in creative practice, e.g., (Saltz 2020), and some psychology research on creativity emphasizes process, e.g., (Glăveanu and Beghetto 2021; Wasserman 2021), but such ideas have not, to my knowledge, made it into algorithms in meaningful ways.

To make the discussion concrete, this paper focuses on algorithms that take a photograph as input and produce a painting as output. Many different types of algorithms for creating digital paintings from input photographs have been developed, including methods based on hand-authored procedures, optimization of brush strokes, learning from example paintings, and learning generative networks. These methods can produce appealing and artistic results. However, they do not handle vague, high-level goals: the style of the output is highly determined by the combination of algorithm, parameters, and inputs used. Indeed, a knowledgable viewer can generally recognize the class of algorithms used, and sometimes the specific algorithm. In contrast, an artist's work can evolve in distinctive and surprising directions.

This paper then proposes possible computational frameworks based on the above observations. I propose to describe vague goals as *underspecified problems,* which may be thought of as optimization problems where high-level choices like style and the specific goals of the painting are part of the search space. In order to model creative processes, the optimization objectives would incorporate perceptual models that can approximate aspects of human judgement of artworks, and the outcomes would depend on both hand-designed exploration processes and numerical optimization. I describe possible ways to design exploratory processes to place brush strokes, incorporating hierarchical task decompositions based on human behaviors.

Existing Computational Painting Frameworks

To focus on a concrete problem domain, this paper discusses stroke-based rendering algorithms that take a photograph as input and produce an image composed of strokes, i.e., curves with color, thickness, and often texture (Hertzmann 2003). The earliest methods were mostly **procedural:** an algorithm defines the steps to create each brush stroke (Haeberli 1990; Litwinowicz 1997; Hertzmann 1998; Zeng et al. 2009; Colton 2012). These procedures embody very specific strategies. For example, Litwinowicz (1997) described a method that places a set of small brush strokes on a jittered grid, sampling colors and orientations from a source image, to achieve an "impressionist" effect (Fig. 1(a)). These methods frequently employ random-number generation to avoid regularity and create variety. Harold Cohen's AARON (1995) is a particularly sophisticated example of hand-authored generative rules for painting, though it is outside the scope of this paper because it does not take a photograph as an input.

Purely procedural methods provide a very limited

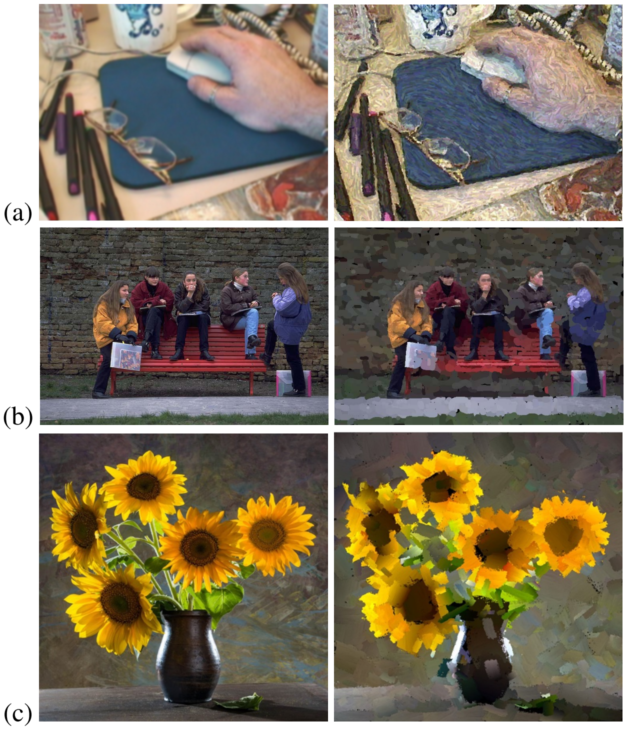


Figure 1: Existing approaches to stroke-based painterly image stylization. (a) A procedural method, where strokes are placed on a jittered grid, drawing color and orientations from a source image (Litwinowicz 1997). The stroke arrangement does not adapt to the source image. (b) An optimization method, allowing strokes to adapt to image content, but with a costly optimization process (Hertzmann 2001). (c) Optimization with differentiable rendering (Zou et al. 2021).

paradigm for understanding painting, since they rely on hard-coded, low-level strategies. Authoring rules for where brush strokes go is very difficult.

This leads to the appeal of **optimization** algorithms (Fig. 1(a)), in which one specifies an objective function for the painting (Hertzmann 2001; Collomosse and Hall 2005; Li et al. 2020; Zou et al. 2021). The objective models the way that an artist may have a goal, e.g., "accurately represent shapes," without requiring the algorithm author to specify a low-level strategy for where brush strokes go. These goals are typically represented with a perceptual image-based loss function, and a generic optimizer is used to optimize the loss, such as gradient descent or evolutionary algorithms. Recent **deep painting** algorithms (Huang, Heng, and Zhou 2019; Jia et al. 2019; Mellor et al. 2019; Nakano 2019; Schaldenbrand and Oh 2021) combine procedural and optimization methods. In these methods, an agent or policy (typically, a Recurrent Neural Network) is trained to optimize an image-based loss. In all of these optimization-based methods, the choice of objective function, its parameters, and any training data, define the artistic style.

Each of these different approaches to painting mirrors different aspects of human creative practices, summarized in Table 1. Specifically, procedural algorithms mimic the use of very specific rules and strategies, e.g., place a jittered grid of strokes; draw big strokes before small strokes. Such rules do not easily adapt to different styles, inputs, or goals. Optimization mimics the search for a high-quality result, e.g., the way a human might iterate over and over on an image until satisfied. Current optimization algorithms correspond to very specific styles; they do not model the way a human might choose a different style for each subject, or even invent new styles along the way. Moreover, they do not model human search strategies, instead they use generic numerical techniques. Deep painting algorithms are optimization algorithms that search for procedures, and thus model how someone might learn to draw, but are also limited to a single style and without explicitly modeling human search.

There are some nuances in the relationship of these approaches. All optimization methods are procedural in the sense that they comprise algorithms and code that generate outputs. But the philosophy for designing optimization algorithms and the philosophy to designing low-level stroke generation procedures are quite different. Likewise, the exploratory processes proposed later in this paper can be thought of as a special case of optimization procedures, but with a different philosophy for how to design them.

A related approach, outside of this paper's focus, uses image processing algorithms without explicit brush strokes (Rosin and Collomosse 2013). Early examples include diffusion (Bangham, Gibson, and Harvey 2003) and physical paint simulation (Curtis et al. 1997). Style transfer methods copy features from example images (Hertzmann et al. 2001; Ramanarayanan and Bala 2007; Gatys, Ecker, and Bethge 2016) to optimize image-based losses. More recently, CLIP-based methods (Radford et al. 2021; Ramesh et al. 2022) optimize images according to a textual prompt rather than an input image (or an input image alone). These methods can randomly select styles or be controlled with style prompts; CLIPDraw (Frans, Soros, and Witkowski 2021) also applies these losses to stroke-based rendering.

Open-ended search. Stanley and Lehman criticize objectives (2015), narrating many evocative examples of human innovation and creativity that did not seem to be the product of goals and objectives. They argue that explicit goals hinder exploration and discovery. Although their ideas have not been used in algorithmic painting, their argument provoked some of the ideas described in this paper.

They propose **open-ended search** as an alternative to optimization. They argue that open-ended search is distinct from optimization because the objective is continually changing. However, operationalizing it as an algorithm distinct from optimization proves elusive. For example, their Novelty Search algorithm (Lehman and Stanley 2011), when applied to path planning, is essentially a variant on the classic goal-directed RRT algorithm (LaValle 1998). Curiosity-driven learning (Pathak et al. 2017) provides another effective computational framework for seeking novelty—also based on an optimization framework—but it is unclear how to apply it to creative tasks.

Human activities	Computer algorithms
Steps, strategy, process	Algorithm, procedure
Goal-directed search	Numerical optimization
Skill learning	Policy optimization, Reinforcement learning
Intrinsically-motivated exploration, creative play	Open-ended search, curiosity-driven learning
Creative problem solving	Underspecified problem solving

Table 1: A possible correspondence between human activities and the computational procedures discussed in this paper. The processes on the right provide models or metaphors to describe the activities on the left. Some of these activities/processes may be complementary, nested, and/or overlapping, e.g., all computational algorithms are procedures. This is not meant to imply equivalence between human behaviors and computational models; as the famous quote by George Box goes: "All models are wrong, but some are useful."

I argue that, in many examples of human innovation, it's not that the innovator lacks a goal or objective, but that the real goal is expressed at a very high level, much more so than in normal optimization problems. This includes many of Stanley and Lehman's (2015) examples. For example, they describe how Elvis Presley's signature sound was not planned, but rather arose simply from playing around in the studio. They use this example to illustrate how "having no objective can lead to the greatest discoveries of all." However, in this example, I argue that Elvis and his band did have an objective: to record a good song. Even though they made unplanned discoveries along the way, these resulted from working toward a high-level goal with an open-ended process, not from aimless exploration. "Open-ended" is a possible description for why someone chooses to make an artwork, but, once that choice is made, the process of making an artwork does have an objective.

Examples from Expert Artists

There appears to be a widespread view that art arises from an artist's specific intent, such expressing an emotion. Computer science discussions of artwork tend to treat the process as fairly linear. For example, to motivate the use of optimization, Durand (2002) writes "Because pictures always have a purpose, producing a picture is essentially an optimization process ... The purpose of the picture can be a message, collaborative work, education, aesthetic, emotions, etc." That is, the artist begins with a goal, and then takes steps toward that goal. I argue that things aren't so simple.

This section provides examples to illustrate two main points. First, **art is typically the product of working toward the vague, high-level goal of making art**. It does not follow a linear path from goals to execution, nor does it come from purely open-ended exploration. Second, **the perceived intent or emotion in a work may often be a *product* of an exploratory process**, rather than its driver. We can often infer intent in a work, but this intent may have come late in the artistic process, if at all.

Pablo Picasso indicated a lack of intent when he said "I don't know in advance what I am going to put on canvas any more than I decide beforehand what colors I am going to use ... Each time I undertake to paint a picture I have a sensation of leaping into space. I never know whether I shall fall on my feet. It is only later that I begin to estimate more exactly the effect of my work." (Read 1960)

Art often does begin with an initial idea or direction, as described by the artist Francis Bacon: "one has an intention, but what really happens comes about in working ... In working, you are following this cloud of sensation in yourself, but don't know what it really is." (Sylvester 1993). That is, his work starts with an initial intention, but it quickly gives way to surprise and discovery. He operates on the high-level goal of making paintings, but the specific intentions of those paintings are not fixed in advance.

Philosopher Nigel Warburton (2003) argues against intent-based definitions of art, citing the above examples from Picasso and Bacon to illustrate "the part played by the unconscious, ..., and the relatively minor role that conscious planning may play in the making of a work of art..." Art critic Jerry Saltz writes "Art is not about understanding and mastery, it's about doing and experience. No one asks what Mozart or Matisse *means*."

Numerous illustrations appear in the recent documentary *Get Back* (Jackson 2021). The documentary follows The Beatles in January 1969 when they were under enormous pressure to write, record, and perform an entirely new album in only a few weeks' time. In one clip[1], Paul McCartney comes into the studio and improvises random sounds on his guitar, until the kernel of a new melody and chorus emerge. We then see The Beatles experimenting with different approaches to refining the song. Ultimately, this song became the hit single "Get Back." The song arose from the high-level goal of making a good song, and then going into the studio and exploring until something emerged.

At one point in this process, The Beatles considered making it a protest song about anti-immigration policies. In this version, the chorus "Get back to where you once belonged," which came from the original jam session, had a totally different meaning than in the final song. Had they released it as a protest song, surely many listeners would have inferred that the song originated from the political message, when in fact the song came before the message.

This example illustrates two kinds of goals in a work. There is the initial, high-level goal ("write a good song"), and the apparent goal or intent of the final song ("protest immigration policies"). As noted by Bacon, there is often also an initial idea or goal that begins the work, but this ini-

[1] `https://youtu.be/rUvZA5AYhB4`

tial goal may be discarded along the way.

Artists carefully consider and develop their artistic processes; process is not merely incidental to outcomes. In the context of the fine art world, Saltz (2020) writes "serious artists tend to develop a kind of creative mechanism—a conceptual approach—that allows them to be led by new ideas and surprise themselves without deviating from their artistic principles." Computer artist Charles Csuri wrote "When I allow myself to play and search in the space of uncertainty, the more creativity becomes a process of discovery. The more childlike and curious I become about this world and space full of objects, the better the outcome" (Greenberger 2022). Painter Gerhard Richter says "I want to end up with a picture that I haven't planned," for which he uses a process that involves chance. "There have been times when this has worried me a great deal, and I've seen this reliance on chance as a shortcoming on my part." But, "it's never blind chance: it's a chance that is always planned, but also always surprising. And I need it in order to carry on, in order to eradicate my mistakes, to destroy what I've worked out wrong, to introduce something different and disruptive. I'm often astonished to find how much better chance is than I am." (Richter, Elger, and Obrist 2009)

Improv theatre is an entire art-form of developing theatre pieces from scratch before a live audience (Johnstone 1979). One of my improv teachers compared it to driving on a dark foggy road in the night, with your headlights illuminating only the road immediately in front of you. All you can do is to keep driving to the next visible spot and continuing from there. You cannot plan, you can only take it one step at a time. Yet, somehow even amateur improv actors can create compelling performances out of nothing. Driving on a foggy night in search of any interesting destination seems like an excellent metaphor for the creative process in general.

The use of creativity exercises (Barry 2019; Brotchie 1993; Parikh and Zitnick 2020) further illustrates the importance of strategy and starting point. Exercises like Exquisite Corpse and automatic drawing can lead to entirely different outcomes each time.

The reader with experience in computer science research may relate to these observations in another way. In many research projects, the goal is to develop new ideas or technologies and publish a paper, while the specific problem being tackled may change along the way during the project. The final paper might look quite different from the initial project idea. It is often said that the most important skill in research is figuring out what problem to work on, and figuring this out is part of the exploration. The distinguished mathematician Michael Atiyah, when asked "How do you select a problem to study?", responded "... I don't think that's the way I work at all. ... I just move around in the mathematical waters ... I have practically never started off with any idea of what I'm going to be doing or where it's going to go. ... I have never started off with a particular goal, except the goal of understanding mathematics." (Minio 2001)

Lessons from Digital Painting

The examples above provide little insight into the specific processes involved. Toward this end, I describe personal experience from my own process of learning to paint digitally. I began digital painting as a hobby in 2019, with the purchase of a new digital tablet and stylus. I had received some training with traditional media many years prior. Now, I painted purely for pleasure, and in the spirit of exploration. But, along the way, I began to recognize specific important features missing from existing approaches to automatic painting algorithms, including the algorithms I had previously developed.

Why might the reader be interested in my own amateur experiences? As I began painting regularly, I observed how my experiences differed from our current computational models for painting. My observations echo the expert examples described in the previous section. But, while many artists have described their own practices at a high level, often these descriptions do not map easily to computational frameworks. Many of my colleagues have tried to develop algorithms by reading art books or talking to artists, only to be frustrated by the seeming impossibility of translating artists' descriptions to algorithms. Here I attempt to relate my experiences to computer science concepts.

For the reader with a computer science background, I hope these stories provide a useful window into artistic experience, targeted to thinking about computational creativity. For the reader with some artistic experience, I hope you may recognize elements of your own experience.

Outcomes are unpredictable

As I began to make my own artwork, I often started with the goal of making my paintings as realistic as possible. Early on, I tried to paint a watercolor of a specific building. After awhile, I became frustrated and disappointed with the painting's progress (Fig. 2); it lacked the detail and precision that I'd wanted. So I switched strategies, adding ink outlines instead, in a way that violated my original goals. The resulting drawing is not in a style that I intended or anticipated, and lacks the realism I'd wanted. Nonetheless, I was happy with the painting and received compliments on it from friends.

A few days later, I decided to try out digital pastels while looking at some flowers on a table at a cafe in front of me. Again, I found the intermediate drawing too messy and again decided to add outlines. I again ended up with a drawing that appeared totally different from my initial goals, but still satisfactory. Moreover, I was surprised how recognizable the style was; like I've seen hundreds or thousands of other drawings in this style. It wouldn't have felt out of place in a hotel room or dentist's office. Perhaps familiarity with this style affected my choices.

The key lesson that I kept relearning over and over is that art comes from a process; I cannot choose the outcome of a new artwork when I begin. I can't even predict it. It's about following the process until I get "something good," not about trying to produce a specific result in a specific style.

Even as I have gained more skill since then, and more ability to control the outcomes of my drawings, still I am surprised again and again by the painting that comes out. My paintings have become more realistic, but still abstracted in surprising (to me) ways. I always start with some idea or

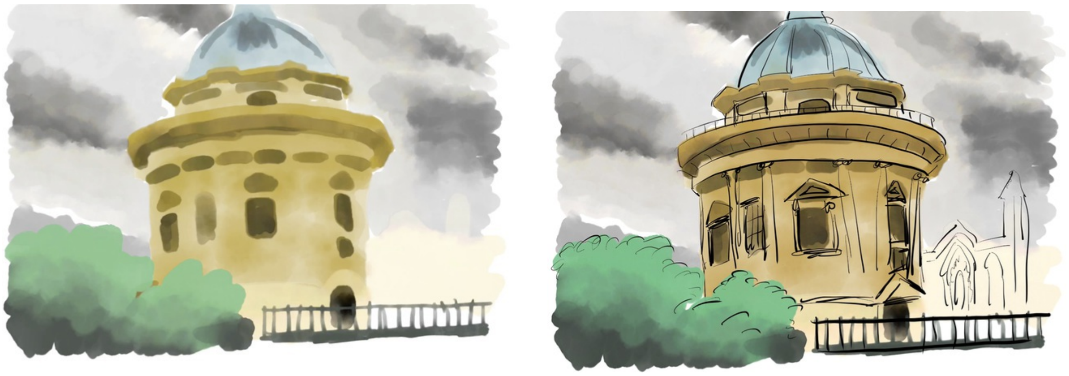


Figure 2: A digital painting of the Radcliffe Camera in Oxford, made in 2019 when I was still getting started with digital drawing tools. I started out intending to make a realistic watercolor of the building. I grew frustrated with watercolor, and decided to switch strategies midway through by adding ink strokes over the watercolor. The final picture didn't meet my initial goals at all—it's not very accurate, and it's not in the style I intended—but I'm still happy with it. Paintings ©2021 Aaron Hertzmann

goal for each painting. But I also have to be ready to shift or abandon that initial idea as the painting emerges.

Of course, highly-trained artists working in more applied domains, e.g., skilled architectural designers, may employ more predictable styles. But, once the style becomes predictable the work becomes less "creative."

The goal of a painting

I started out solely painting from life, without photography, in order to better develop my skills. At some point, I began taking photos along with each painting, so that I would have a reference in case I wanted to keeping working on the painting later.

And I noticed two things. First, putting the photograph next to the painting made the painting look "wrong." Side-by-side, I could see all the technical flaws in the painting.

Second, I didn't care. I still liked the painting more.

This seemed like a contradiction: I wanted to make images look as real as possible, yet, I didn't want to duplicate photographs, and I was quite happy when my paintings were not photographic. Indeed, had I truly sought photorealism, I could just take photographs.

These thoughts led me to the following realization, which seems obvious, even vacuous, but is really quite important: **The goal of painting is to make a good picture.**

What is a "good" picture? There are lots of ways that a picture can be "good." For me, "good" does not mean photorealistic, or even accurate. It means that I like looking at it, that other people like looking at it. Depicting reality can be part of it. Or perhaps the picture conveys something about my subjective experience. Or maybe it just looks nice.

I always start out with some kind of goal or idea for a painting. But, my goals might change along the way, and, ultimately, I seek only a painting that somehow achieves something. Along the way, I assess my progress—and whether I am done and can stop—looking at the painting and evaluating it, and where it may need improvement, using my own judgement as to whether the painting is as good as I can make it, and what parts I can try improving. In these assessments I am simultaneously watching the painting evolve and discovering what its "goals" might be.

I sometimes sense a jarring disconnect between the way that others (quite understandably) interpret my paintings, versus the actual history of how those paintings evolved. One friend commented that my painting allowed her to see through my eyes, yet I thought it was poor depiction of reality. Another friend commented that I'd done a good job of capturing the lighting in a scene. I didn't think the painting conveyed my actual experience well—it's just that lighting I painted looked good anyway.

Dependence on choices and content

In optimization algorithms, the objective and the constraints are meant to determine the outcomes, and any dependence on initialization or parameterization is viewed as a shortcoming. Yet, if the artist's goals were all that mattered, then artistic process would be no more than a matter of developing technical skill. In my own experience, initial choices of strategy, media, and process are paramount to the outcome. Existing algorithms, typically treat an image's "style" and its "content" as independent, e.g., (Hertzmann et al. 2001; Gatys, Ecker, and Bethge 2016). Yet, often the style that emerges is very much a function of the scene I'm trying to depict.

At each stage of a painting, I have many choices to make. Which media should I use—solid brushes, oil simulation, watercolor simulation, or something else? Should I try a new brush I haven't tried before? Should I draft an outline first, or just start drawing the first object that catches my eye? Should I start drawing the background or foreground first? And so on.

I found that every single one of these choices has a transformative effect on the resulting painting. Something so seemingly inconsequential as starting with a dark background would lead to a painting with a completely different character than had I begun with white background. In some cases, these choices were made randomly or by accident; I only started drawing with pencil because of a time when

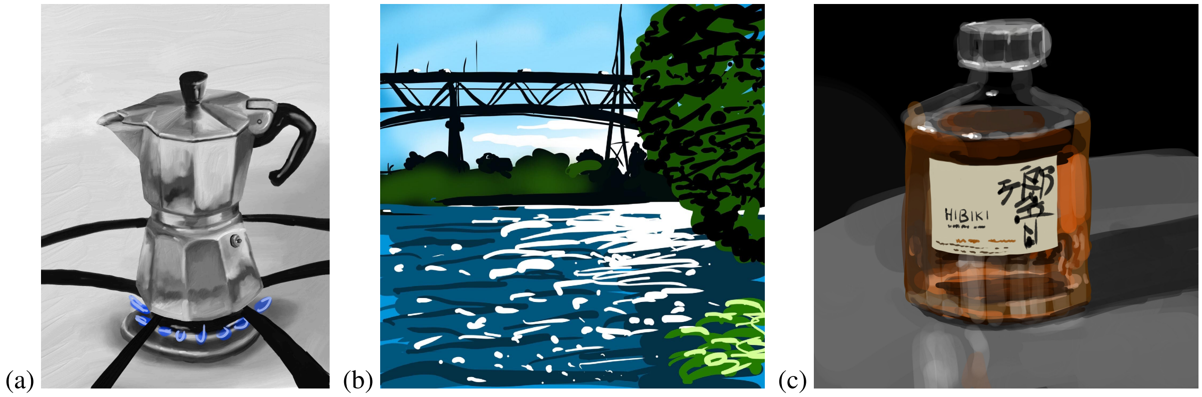


Figure 3: Three examples of digital drawings in which the subject determined the style I ended up using. (a) I used simulated oil paint to depict shading variations. (b) I used solid-color strokes, since the water could be clustered into three distinct colors. (c) I used semi-transparent, texture-less strokes to illustrate transparency and refraction. While each of these subjects could have been drawn in any of these styles, the results would have been very different, and it would have been much more difficult to achieve a satisfying result. Paintings ©2021 Aaron Hertzmann

I forgot to change the brush from the default, and quickly discovered that I loved it. While it is hypothetically possible to steer a painting far from where it began, it is rarely worthwhile, since it takes much more time and may produce a painting which is more "stale."

Often my initial choices about media and style will be a function of the scene I'm trying to depict. For example, in Figure 3(a), the real scene involved smooth tonal gradients, and so I chose simulated oil paint, which allows me to use blending with wet-in-wet painting. In contrast, in Figure 3(b), the scene involved stark late afternoon lighting, where the tones off the water could be clustered into a few colors, due to the Fresnel effect reflections on the water: one for bright sunlight reflections, one for sky reflections, and one for refraction; the bridge appeared silhouetted against the sky, so a solid black was sufficient. Hence, I chose an opaque, solid color brush for the water and the bridge, with no need for blending. In Figure 3(c), the object had complex transparency and reflection, so I chose semi-transparent strokes together with layering as provided in the drawing app.

In principle, I could choose any strategy for any scene. I have tried drawing complex architectural scenes without an initial sketch; they often come off loose and sloppy. I could have depicted the transparent bottle with a single layer of oil paint strokes. This would have been far more difficult to paint, most likely doing a poorer job at capturing the transparency. And sometimes these alternative approaches produce results that are appealing in other ways; it is truly hard to claim that one approach is intrinsically better than another.

Figure 4 shows four paintings painted at roughly the same time and location, with different techniques and very different outcomes.

In short, the **the style of an image arises both from the subject of the scene and the techniques chosen along the way,** rather than the starting with a desired style and goals.

Intuitions and Conscious Choices

So how do all of these choices get made? Much of learning to paint is about developing intuitions. In my initial attempts, sometimes I would consciously decide to try a new approach or technique. Some of these experiments felt like failures, others felt like unexpected successes. From this experience, I have developed intuitions about which choices to make. Considering a new subject, I may consciously choose whether or not to begin sketching an outline, or simply to start drawing the nearest object. I might consider how sloppy the subject might look without the sketched outline, versus the extra time it would take to do so, and the danger of losing spontaneity. Or, if I'm in a rush, I'll pick one without too much deliberation.

At each stage, these is a question of what to work on next. Refine details in one object? Adjust the overall arrangement? Fix the background? There are countless options at each stage, and conscious deliberation would take forever. One skill is to look at the current state of the painting and select the next element to work on.

At times, I do stop and stare at the painting, sometimes comparing it to the real subject. It can take time to get a sense for what is working in the painting and what to improve.

At some point, I tend to transition from exploration to refinement, improving little details and fixing little flaws. One could say that refinement happens once the style and content of the painting have emerged.

One of the biggest problems is deciding when to stop. I particularly struggled with this when I used to use physical oil paint and watercolor. At some point I'd be trying to refine a piece, and each new change would make it worse. Digital tools and "undo" are more forgiving, but it can still be hard to recognize the point at which there is no benefit to continuing to work on a painting. According to one quote, attributed to many artists at different times: "A work of art is never completed, only abandoned."

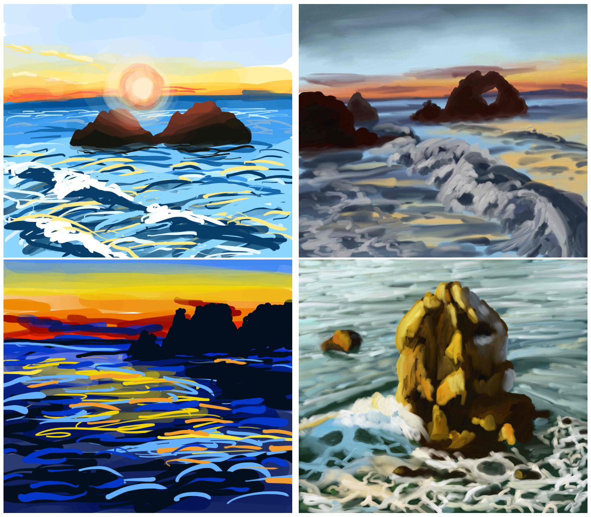


Figure 4: Four paintings made at roughly the same time in the same spot, illustrating how different choices of media and technique can produce very different outcomes. I painted the first three quickly en plein air, and the fourth from a photograph later on. Each one surprised me when completed. Paintings ©2021 Aaron Hertzmann

Intuition or Impulse. Instinctive choices often feel magical, and there is considerable mythology around the idea of "artistic insight." To what extent are these seemingly-ineffable choices about intuitions, random impulse, or some other "subconscious force"?

In the language of Kahneman and Tversky (2011), intuitions correspond to System 1 and conscious choice to System 2, which, in machine learning, can be associated with supervised learning and conceptual reasoning, respectively (Bengio 2019). In the extreme version of this view, intuitions are optimizations that allow one make decisions from experience without further cognition.

One can also associate intuitions and cognition with Primary and Secondary Processes, which is hypothesized in psychology to be crucial to creativity. As summarized by Runco (2014), primary processes reflect "impulse, libido, and uncensored thoughts and feelings," while secondary processes are "purposeful, rational, and guided by conventional restraints." Several studies describe creative work as a "magic synthesis," a collaboration of primary and secondary processes.

Toward Algorithms

How can we devise algorithms that capture some of the above observations? I now propose a framework with two components: a formulation of vague, high-level goals, and a structured exploratory search process for finding outputs that satisfy the goals. Following the observations in the previous sections, both elements are essential components.

Underspecified Problems

In conventional painting optimizations, the *style* of a painting is largely fixed in advanced by user-set loss functions and their weights, the type of brush strokes (e.g., fixed stroke sizes or lengths, texture or paint simulation, etc.), limits on the number of strokes, style training images, and so on.

I first propose to model painting as an *underspecified problem.* I define an underspecified problem as a problem for which there are many valid solutions that correspond to different high-level choices, such as different choices of media, abstraction, style, emphasis, exaggeration/distortion, apparent intent, and so on; any of these would be considered a "valid" solution to the problem. An output that satisfies the goals of the problem might be a style adapted to fit the subject well; it could be a new style. An underspecified problem could be phrased as, for example, "make me a nice painting of a tree in any style," modeled as optimizing a painting of that tree for human perception and aesthetic appreciation. For specific applications, there may be other goals orthogonal to the underspecified problem, for example, more precise depictions of shape, or style matching a specific user's preferences.

In research, underspecified problems might include "prove an interesting theorem" or "write a publishable paper."

Underspecified problems would likely include subjective or hard-to-quantify goals, such as aesthetic beauty and visual novelty. This requires developing a *perceptual model* that models the artist's judgment of their own work-in-progress (Moruzzi 2021), which provides the objective function for optimization, incorporation notions of aesthetics, scene perception, and novelty. Developing such a model is a "grand challenge" problem, but one for which incremental progress could still lead to useful algorithms. That is, it does not need to truly capture human perception to lead to useful painting algorithms, just as the Creative Adversarial Network (Elgammal et al. 2017) produces interesting styles with only a limited model. Curiosity-driven learning (Pathak et al. 2017) presents possible insights for modeling visual novelty.

Rather than building such an artificial "critic," one could use human judgements. This human-in-the-loop approach has also been explored extensively in other contexts, for collaborative artistic exploration (Draves 2005; Sims 1991; Secretan et al. 2011; Klingemann and others 2021) and for exploratory user interfaces, e.g., (Koyama, Sato, and Goto 2020; Marks et al. 1997). Including humans in the loop limits an approach's usefulness and its ability to model real processes. But these human judgements could be used to bootstrap or improve the critic.

Artistic Exploration and Search Algorithms

It is not sufficient to have a perfect model of a human viewer's perception and judgement of a work, even if such a thing were possible. Fortunately, it is not necessary that the model be perfect either. The exploration and search algorithm is also crucial, and it can make up for limitations of the perceptual model. The search algorithm's goal is to find good solutions to the underspecified problem; moreover, the design of the exploratory search process can help determine the space of styles as well.

There are several aspects that such an exploratory search procedure would likely have. All of these are about making choices at each step: choosing strategies, choosing media, choosing where individual strokes go, and so on. Generic optimization algorithms as used in existing methods are not sufficient.

Explicit task decomposition. When starting a painting, one may choose a strategy, e.g., start with an outline sketch, start drawing the first object, start drawing the background, and so on. Within each strategy one has a set of sub-tasks, e.g., drawing specific objects, refining shape, or color, or shading, or each together, evaluating the current painting and deciding whether to stop, etc.

Algorithmically, this corresponds to hierarchical task decomposition (Lu et al. 2021). A painting algorithm could be represented as a loop of selecting a current strategy, and, within this strategy, selecting the next task to perform, and then performing it for some duration, and then selecting the next task. While the parameters could be learned in some cases, the design of these classes of strategies and tasks would likely be done by the algorithm designer.

Transition from Exploration to Optimization. The search should also be guided by the underspecified loss function. At early stages of the painting, the loss may play a small role, whereas later steps should largely be refinement, similar to conventional optimization. In other words, the algorithm may behave much like a hand-designed procedural algorithm early in the process, and more like an optimization algorithm later on.

Randomness when making choices plays a key role in producing unexpected outputs, and randomness is more important early in the process. Randomly selecting a strategy early on could lead to an entirely different output, whereas randomness in stroke placement toward the end of the process may only provide a bit of stroke jittering.

Learned vs. procedural decision-making. How does the system make these choices? The simplest answer is to assess the change to the painting according to the underspecified loss, and randomly pick from the high-scoring options. In this way, the loss can guide decision-making. A second answer is to procedurally author rules for the early stages of the process. A more intriguing and complex approach would be to use existing deep learning frameworks to learn some parameters of these decision-making steps, for example, by having users rate outputs of the system and allowing the system to learn over time. This would be distinct from the existing systems by the use of the hierarchical task decomposition and the underspecified loss. Moreover, it could be intriguing to watch the system learn and see its styles evolve.

Conclusion

The central thesis of this paper is that many creative practices can be described as satisfying vague, high-level goals through exploratory search processes, and that we can attempt to model these practices computationally. Building such a computational formulation includes difficult "grand challenge" problems, such as the problem of sufficiently approximating how a viewer perceives a new painting.

The idea of goal-directed exploration overlaps with many ideas of creativity and open-ended search (Stanley and Lehman 2015), such as quality and novelty. But it is more specific: it implies the existence of a high-level goal (produce an output image), and it suggests the existence of a mathematical formulation (the language of optimization). Open-endedness and curiosity-driven (Pathak et al. 2017) are good descriptions of why we might choose to paint at all, whereas the framework described in this paper describes the act of making a specific painting.

It is hard to know which aspects of human creativity are "fundamental," and which are secondary effects/epiphenomena (Jordanous 2012). Which attributes of creative people, products, and behaviors are important? For example, it has often been asserted that left-handed people are more likely to be creative (van der Feen et al. 2020). However, forming a research program around left-handed robots does not sound fruitful. This paper points out attributes of creative practice that, to my knowledge, have not been deeply explored in the creativity literature and may be important candidates, in addition to, or instead of, concepts like novelty and effectiveness (Boden 1998; Jordanous 2012; Runco and Jaeger 2012).

Many definitions of creativity focus on the qualities of the output, including both in the psychology literature, e.g., (Runco and Jaeger 2012), and in computation, e.g., (Boden 1998; Colton and Wiggins 2012). Yet, in some cases, very simple rules can produce results that are judged as "creative" by experts, e.g., (Goldenberg, Mazursky, and Solomon 1999). Instead, some researchers have argued for understanding the process itself in defining creativity (Glăveanu and Beghetto 2021) and evaluating it (Colton 2008; Moruzzi 2021).

This paper focuses on building systems inspired by human creativity, and, toward this end, we likewise argue that it is not sufficient to consider losses and evaluations, but to carefully formulate the processes by which artifacts are produced when designing these systems.

Author Contributions

AH ideated and wrote the paper alone.

Acknowledgements

Thanks to Josh Davis, Laura Herman, and Manuel Ladron De Guevara.

References

Bangham, J. A.; Gibson, S. E.; and Harvey, R. W. 2003. The art of scale-space. In *BMVC*, 1–10.

Barry, L. 2019. *Making Comics*. Drawn & Quarterly.

Bengio, Y. 2019. From system 1 deep learning to system 2 deep learning. NeurIPS keynote talk. `https://www.youtube.com/watch?v=FtUbMG3rlFs`.

Boden, M. A. 1998. Creativity and artificial intelligence. *Artificial intelligence* 103(1-2):347–356.

Brotchie, A. 1993. *Surrealist Games*. Redstone Press.

Cohen, H. 1995. The further exploits of AARON, painter. *Stanford Humanities Review* 4(2).

Collomosse, J. P., and Hall, P. M. 2005. Genetic paint: A search for salient paintings. In *Workshops on Applications of Evolutionary Computation*.

Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier? In *Proc. ECAI*, volume 12, 21–26. Montpelier.

Colton, S. 2008. Creativity versus the perception of creativity in computational systems. In *AAAI spring symposium: creative intelligent systems*.

Colton, S. 2012. The Painting Fool: Stories from Building an Automated Painter. In *Proc. Computers and Creativity*.

Curtis, C. J.; Anderson, S. E.; Seims, J. E.; Fleischer, K. W.; and Salesin, D. H. 1997. Computer-generated watercolor. In *Proc. SIGGRAPH*, 421–430.

Draves, S. 2005. The Electric Sheep Screen-Saver: A Case Study in Aesthetic Evolution. In *Proc. EvoWorkshops*.

Durand, F. 2002. An invitation to discuss computer depiction. In *Proc. NPAR*.

Elgammal, A.; Liu, B.; Elhoseiny, M.; and Mazzone, M. 2017. CAN: Creative Adversarial Networks Generating "Art" by Learning About Styles and Deviating from Style Norms. In *Proc. Int. Conf. on Computational Creativity*.

Frans, K.; Soros, L.; and Witkowski, O. 2021. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. *arXiv preprint arXiv:2106.14843*.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2016. Image Style Transfer Using Convolutional Neural Networks. In *Proc. Computer Vision and Pattern Recognition*.

Glăveanu, V. P., and Beghetto, R. A. 2021. Creative experience: A non-standard definition of creativity. *Creativity Research Journal* 33(2):75–80.

Goldenberg, J.; Mazursky, D.; and Solomon, S. 1999. Creative sparks. *Science* 285(5433).

Greenberger, A. 2022. Charles csuri, 'father of computer art,' dies at 99. `https://www.artnews.com/art-news/news/charles-csuri-dead-1234621107/`.

Haeberli, P. 1990. Paint By Numbers: Abstract Image Representations. In *Proc. SIGGRAPH*.

Hertzmann, A.; Jacobs, C. E.; Oliver, N.; Curless, B.; and Salesin, D. H. 2001. Image Analogies. In *Proc. SIGGRAPH*.

Hertzmann, A. 1998. Painterly Rendering with Curved Brush Strokes of Multiple Sizes. In *Proc. SIGGRAPH*.

Hertzmann, A. 2001. Paint by relaxation. In *Proc. CGI*.

Hertzmann, A. 2003. A Survey of Stroke-Based Rendering. *IEEE Computer Graphics & Applications* 23(4).

Huang, Z.; Heng, W.; and Zhou, S. 2019. Learning to paint with model-based deep reinforcement learning. In *Proc. ICCV*.

Jackson, P. 2021. The Beatles: Get Back. [motion picture].

Jia, B.; Brandt, J.; Mech, R.; Kim, B.; and Manocha, D. 2019. Lpaintb: Learning to paint from self-supervision. In *Proc. Pacific Graphics*.

Johnstone, K. 1979. *Impro: Improvisation and the Theatre*. Faber and Faber.

Jordanous, A. 2012. A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. *Cognitive Computation* 4(3):246–279.

Kahneman, D. 2011. *Thinking: Fast and Slow*. Farrar, Straus and Giroux.

Klingemann, M., et al. 2021. Botto, decentralized autonomous artist. `https://botto.com/`.

Koyama, Y.; Sato, I.; and Goto, M. 2020. Sequential gallery for interactive visual design optimization. *ACM Trans. Graph.* 39(4).

LaValle, S. M. 1998. Rapidly-exploring random trees: A new tool for path planning. Technical Report TR 98-11, Computer Science Dept., Iowa State University.

Lehman, J., and Stanley, K. O. 2011. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation* 19(2):189–223.

Li, T.-M.; Lukáč, M.; Michaël, G.; and Ragan-Kelley, J. 2020. Differentiable vector graphics rasterization for editing and learning. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 39(6):193:1–193:15.

Litwinowicz, P. 1997. Processing Images and Video for an Impressionist Effect. In *Proc. SIGGRAPH*.

Lu, Y.; Shen, Y.; Zhou, S.; Courville, A.; Tenenbaum, J. B.; and Gan, C. 2021. Learning task decomposition with ordred memory policy network. In *Proc. ICLR*.

Marks, J.; Andalman, B.; Beardsley, P. A.; Freeman, W.; Gibson, S.; Hodgins, J.; Kang, T.; Mirtich, B.; Pfister, H.; Ruml, W.; Ryall, K.; Seims, J.; and Shieber, S. 1997. Design galleries: A general approach to setting parameters for computer graphics and animation. In *Proc. SIGGRAPH*, 389–400.

Mellor, J. F. J.; Park, E.; Ganin, Y.; Babuschkin, I.; Kulkarni, T.; Rosenbaum, D.; Ballard, A.; Weber, T.; Vinyals, O.; and Eslami, S. M. A. 2019. Unsupervised doodling and painting with improved spiral. arXiv:1910.01007.

Minio, R. 2001. An Interview with Michael Atiyah. *Mathematical Conversations*. `https://doi.org/10.1007/978-1-4613-0195-0_2`.

Moruzzi, C. 2021. Measuring creativity: an account of natural and artificial creativity. *Euro Jnl Phil Sci* 1(11).

Nakano, R. 2019. Neural painters: A learned differentiable constraint for generating brushstroke paintings. arXiv:1904.08410.

Parikh, D., and Zitnick, C. L. 2020. Exploring crowd co-creation scenarios for sketches. In *Proc. ICCC*.

Pathak, D.; Agrawal, P.; Efros, A. A.; and Darrell, T. 2017. Curiosity-driven exploration by self-supervised prediction. In *ICML*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning transferable visual models from natural language supervision. In *Proc. ICML*.

Ramanarayanan, G., and Bala, K. 2007. Constrained texture synthesis via energy minimization. *IEEE Transactions on Visualization and Computer Graphics* 13(1):167–178.

Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical text-conditional image generation with clip latents. arxiv.org/abs/2204.06125.

Read, H. 1960. *Art Now: an Introduction to the Theory of Modern Painting and Sculpture*. Faber & Faber.

Richter, G.; Elger, D.; and Obrist, H. U. 2009. *Gerhard Richter — Text: Writing, Interviews and Letters 1961–2007*. London: Thames & Hudson.

Rosin, P., and Collomosse, J., eds. 2013. *Image and Video-Based Artistic Stylization*. Springer.

Runco, M. A., and Jaeger, G. J. 2012. The standard definition of creativity. *Creativity Research Journal* 24(1):92–96.

Runco, M. A. 2014. *Creativity*. Elsevier, 2nd edition.

Saltz, J. 2020. *How To Be An Artist*. Riverhead Books.

Schaldenbrand, P., and Oh, J. 2021. Content masked loss: Human-like brush stroke planning in a reinforcement learning painting agent. In *Proc. AAAI*.

Secretan, J.; Beato, N.; D'Ambrosio, D. B.; Rodriguez, A.; Campbell, A.; Folsom-Kovarik, J. T.; and Stanley, K. O. 2011. Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary computation* 19(3):373–403.

Sims, K. 1991. Artificial evolution for computer graphics. In *Proc. SIGGRAPH*.

Stanley, K., and Lehman, J. 2015. *Why Greatness Cannot Be Planned: The Myth of the Objective*. Springer.

Sylvester, D. 1993. *Interviews with Francis Bacon*. Thames & Hudson.

van der Feen, F. E.; Zickert, N.; Groothuis, T. G.; and Geuze, R. H. 2020. Does hand skill asymmetry relate to creativity, developmental and health issues and aggression as markers of fitness? *Laterality* 25(1):53–86. PMID: 31117903.

Warburton, N. 2003. *The Art Question*. Routledge.

Wasserman, E. A. 2021. *As If By Design: How Creative Behaviors Really Evolve*. Cambridge University Press.

Zeng, K.; Zhao, M.; Xiong, C.; and Zhu, S.-C. 2009. From image parsing to painterly rendering. *ACM Trans. Graph.* 29(1):2:1–2:11.

Zou, Z.; Shi, T.; Qiu, S.; Yuan, Y.; and Shi, Z. 2021. Stylized neural painting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15689–15698.

CLIP-CLOP: CLIP-Guided Collage and Photomontage

Piotr Mirowski, Dylan Banarse, Mateusz Malinowski, Simon Osindero, Chrisantha Fernando
DeepMind, London, UK
{piotrmirowski, dylski, mateuszm, osindero, chrisantha} @ deepmind.com

Abstract

The unabated mystique of large-scale neural networks, such as the CLIP dual image-and-text encoder, popularized automatically generated art. Increasingly more sophisticated generators enhanced the artworks' realism and visual appearance, and creative prompt engineering enabled stylistic expression. Guided by an artist-in-the-loop ideal, we design a gradient-based generator to produce collages. It requires the human artist to curate libraries of image patches and to describe (with prompts) the whole image composition, with the option to manually adjust the patches' positions during generation, thereby allowing humans to reclaim some control of the process and achieve greater creative freedom. We explore the aesthetic potentials of high-resolution collages, and provide an open-source Google Colab as an artistic tool.

Introduction

A *collage*, from the French *coller*, is "a composite image made by sticking newspaper cuttings, photographs, and other printed images onto a flat surface, often combined with paint" (Zaczek and Actor 2008). *Photomontage* extends collage by manipulating and compositing photographs (Ades 1976). The origins of collage can be traced to the invention of paper in China, and *photo-collage* was a social pastime for the Victorian upper-class (National Galleries of Scotland 2019), before Cubists Pablo Picasso and Georges Braque made collage into an art form (Zaczek and Actor 2008; Greenberg 1958).

In this paper, we formalize collage as a picture produced by optimizing affine spatial and color transformations of patches, where patches are manually selected, and then automatically sampled, moved around, recolored, and superimposed. We design a gradient-based *Collage Generator* consisting of differentiable spatial and color transformations of patches followed by transparent or opaque superposition.

The *Collage Generator* optimizes such transformations guided by a dual text-and-image encoder (Liu, Gong, and others 2021), like the popular CLIP model from OpenAI (Radford, Wook Kim, and others 2021), pre-trained on large datasets of captioned images collected on the internet (hence incorporating various cultural biases). Intuitively, the dual encoder computes a score for the match between a textual *prompt* and the resulting *collage image*. Therefore, it acts as an AI-based *Critic* assessing the "quality" of the artwork given its description. Large-scale dual encoders exhibit some degree of semantic compositionality, as they allow novel combinations of phrases or images, and handle such visual concepts as color, texture, shape, object relations, perspective and "style", to guide a generator to create remarkably convincing images.

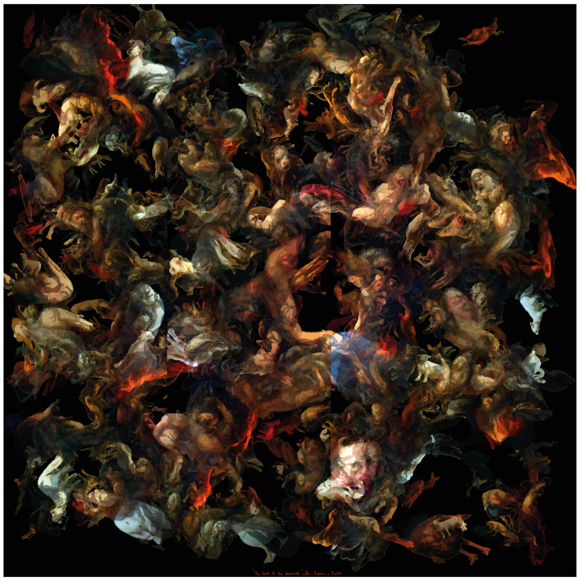


Figure 1: *The Fall of the Damned after Rubens and Eaton*. High-resolution collage of image patches of animals (Fig.7), optimized hierarchically with 3x3 overlapping CLIP critics.

Computational artists like Ryan Murdock, Katherine Crowson and Mario Klingemann have investigated various neural generators, including Generative Adversarial Networks (Brock, Donahue, and Simonyan 2018; Esser, Rombach, and Ommer 2021), Diffusion models (Dhariwal and Nichol 2021), evolution strategies on colored shapes (Tian and Ha 2021) or evolution-based Neural Visual Grammars (Fernando, Eslami, and others 2021); each producing distinctive aesthetics in tandem with the CLIP critic. In Spring 2021, open-source Google Colabs allowing practitioners to combine VQGAN generators with CLIP critics (Crowson et al. 2022) greatly popularised the technique. More recent methods that rely on Latent Diffusion conditioning or direct prediction of CLIP image embeddings manage to forgo the lengthy CLIP critic iterations and allow considerably faster and higher quality text-to-image generation (Rombach et al. 2021; Ramesh et al. 2022).

Our system is more interpretable, as it merely optimises color and affine transformations of hand-selected patches, instead of optimizing latent variables that condition a neural pixel image generator. Since the *Collage Generator* operates on collections of identifiable patches, we can let the user intervene during the optimization, manually adjusting the arrangement (shift, scale and rotation) of individual patches, for additional human-in-the-loop guidance.

Our work extends differentiable scalable vector graphics used in CLIPDraw (Frans, Soros, and Witkowski 2021), substituting strokes with patches. We experiment with various rendering methods for superimposing patches in a learnable way. We can also combine multiple critic evaluations on overlapping regions of a larger image to produce high-resolution[1] and detailed collages, allowing the artist control over the composition of the artwork. We call our system *CLIP-CLOP* (loosely CLIP-guided COLlage and Photomontage) and open-source its Google Colab code.

CLIP-CLOP also builds upon extensive prior work in computational creativity. Automated collage generation (Krzeczkowska et al. 2010) in the *The Painting Fool* (Colton 2008) employed keyword retrieval-based methods to make thematic news-based juxtapositions of images. Optimisation methods were used for spatial and colour transformations of image *cutouts* to assemble “Arcimboldo”-like collages that match a target image (Huang, Zhang, and Zhang 2011). Semantic composition of patches were conceptualised as relative positions between image cutouts in (Breault et al. 2013), and then explored as juxtaposition, replacement and fusion of images in (Xiao, Linkola, and others 2015) or as visual blending of emoji pictograms in (Cunha, Martins, and Machado 2018). CLIP-CLOP combines all above aspects with differentiable transformers and a CLIP critic for textual prompt-driven image generation.

CLIP-CLOP is directly inspired by art theory. First, CLIP-CLOP arranges disparate collections of textured patches into new images, just like collage techniques enabled Cubist artists to exploit ambiguities arising from the shapes and perspectives of patches (Zaczek and Actor 2008). Second, Hans Arp’s *Collage With Squares Arranged according to the Law of Chance* (1916-1917)[2], is a precursor to CLIP-CLOP’s random initialization and optimization of patches, with optional manual adjustment, and an illustration of our human-in-the-loop approach to procedural art generation. We believe that allowing patch choice gives the artist more control than the mere combination of prompt engineering with critic-guided generators (Radford, Wook Kim, and others 2021) and situates CLIP-CLOP with recent human-in-the-loop works.

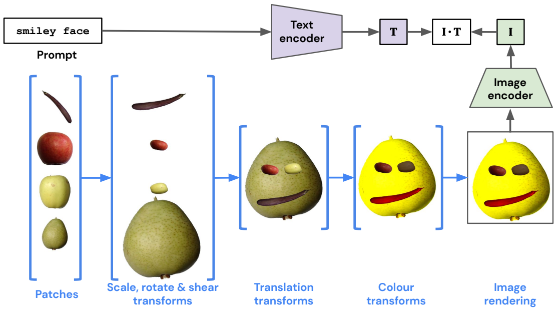


Figure 2: Architecture of the generative collage algorithm.

Algorithm

Like in traditional *collage*, the artist prepares a collection of N image patches. CLIP-CLOP randomly initialises N RGB color transformation vectors and N affine transformation matrices, one for each patch, to randomly color and disperse them onto a canvas. These patch transformations, as well as the patch superposition and image rendering method, constitute the *Collage Generator*. A forward pass through this *Collage Generator* applies transformations to each patch and then combines patches by superimposing their RGB images onto a blank canvas. The resulting image is then evaluated by the *Critic* (dual image-and-text encoder) and matched to one or several user-defined textual prompts. The automated optimization loop between the parameters of the *Collage Generator* and the *Critic*’s evaluation is illustrated on Figure 2. An optional evolution-based optimization can be applied to the set of image patches.

Preparation of Image Patches

We leave that crucial curation process to the artist, as it uniquely defines the artwork, and offer only basic tools for preparing image datasets.

CLIP-CLOP takes as input a list of N 4-channel (RGB plus alpha) image arrays. The contour of each patch is specified by the alpha channel, which can be semi-transparent. We explored manual image segmentation using photo editing software, automated flood-filling from four corners of each patch image (when those images are adequately centered photographs of objects on a plain background) and computer vision-based image segmentation of photographs over cluttered backgrounds.

Collage Generator

The *Collage Generator* is composed of color transformation, spatial affine transformation of each patch, and patch superposition, three operations that are differentiable, allowing gradient-based optimization.

Color Transformation Each given image patch is assigned three color multipliers, for the red, green and blue channels; changing those coefficients with values smaller

[1] Open-source code and examples of high-resolution images: `https://github.com/deepmind/arnheim`

[2] Museum of Modern Art, New York: `https://www.moma.org/collection/works/37013`

than 1 uniformly changes the patch's color. These parameters are optimized during training.

Spatial Transformations Similarly, each image patch is assigned six numbers, for X and Y translation, rotation, scale, squeeze and shear along the X axis. The two-dimensional affine transformations are expressed as 3 × 3 translation, rotation and scale matrices, and are applied to the image pixel 2D coordinates. The resulting affine-transformed (rotated, scaled and translated) grids of pixel coordinates are then used to interpolate the patch. Similarly, these affine transform parameters are optimized during collage generation.

Differentiable Rendering Collage artworks typically superimpose opaque scraps of paper cuttings, assuming a partial ordering – which scrap is on top of another. Yet in our case of differentiable rendering, a completely opaque superposition of patches compromises the learnability of the collage because we cannot propagate gradients through occluded patches.

We thus investigated two alternatives. The first one, *transparency*, simply adds the RGB values of all patches. A variation of *transparency*, called *masked transparency*, sums RGB values of only non-masked parts of patches (i.e., where alpha channel values are strictly greater than 0) and normalizes each pixel value by the sum of masks at that position.

The second one, called *opacity*, replaces the opaque, ordered superposition of patches by a differentiable approximation, consisting of a weighted sum of patches with learnable patch weights. Specifically, each patch is given an order parameter. For each patch and pixel coordinate, we compute the weight by multiplying the order of the patch by the mask at that pixel. The resulting image is a weighted sum of patches. Again, patch order parameters are optimized during collage generation.

Figure 3 shows the effect of various rendering methods. Note that *opacity* does not always fully occlude all image patches but does more so than *transparency* and *masked transparency*, and that (unless the RGB range is allowed to be negative) *transparency* can result in saturated (close to white) image parts (visible on left image in some of the coral tentacles).

Achieving High Resolution CLIP-CLOP's advantage is that it can produce collages at any resolution. During optimization, we use down-sampled patches, as CLIP is restricted to 224 x 224 images; for the final rendering, the same spatial transformations are applied to the original high resolution patches.

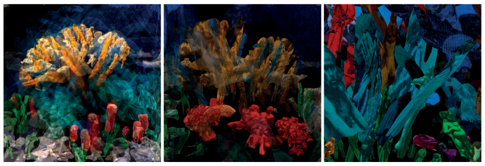


Figure 3: Rendering methods on prompt *underwater coral*. Left to right: transparency, masked transparency, opacity.

Critic

By a crude analogy to an art critic, who interprets and evaluates a piece of art, we use a compatibility function – called *Critic* – between the collage and the textual prompts. Intuitively, the higher the score given by the *Critic* function, the better the fit between textual and visual inputs. At each step, *Collage Generator* produces parameterized transformations of patches rendered on the canvas and generates a new collage proposal. Next, we use CLIP (Radford, Wook Kim, and others 2021) – a large-scale model, trained on 400 million image-text pairs – as *Critic*, with an encoder that extracts image features from the collage and text features from the prompt. These features are matched against each other to give a compatibility score.

During training, that score is optimised by stochastic gradient ascent and backpropagated through the image to the *Collage Generator* to optimize the patches' transformations parameters.

Semantic Composition Many generative approaches produce images with single semantics, i.e., evaluated globally with one *Critic*. To achieve a higher level of compositionality, we divide the image into 3×3 overlapping local regions, each evaluated by a different *Critic* and prompt. A tenth *Critic* evaluates the whole collage globally (with reduced resolution). Figure 4 illustrates how one could decompose "landscape" using prompts: "sky with sun", "sky" and "sky with moon", "trees", etc.

Moreover, the same procedure allows to increase the resolution of the final collage. With 3 × 3 regions, we can produce 448 × 448 images instead of 224 × 224, typical of approaches that use CLIP. In our work, we experiment with parallel *Critic* evaluations and less memory consuming but slower serial evaluations. We use either arithmetic or harmonic mean of all individual *Critic* losses.

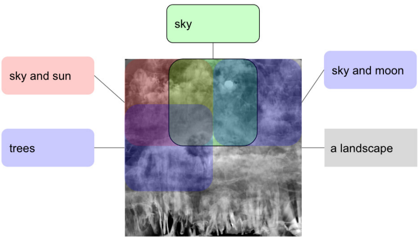


Figure 4: Using multiple overlapping CLIP evaluators with different prompts allows greater control over composition and higher resolution collages.

Evolution of Image Patches Gradients enable existing patches to be manipulated but do not provide a signal for exchanging one patch for another. To support this, we optimize a population of 2 to 10 randomly initialized collages and apply a step of evolution (microbial genetic algorithm (Harvey 2009)) every 100 gradient descent steps. The scores of two random *Collage Generators* are compared and the loser is overwritten with the winner, with random mutations involving swapping a random patch for another or small Gaussian noise added to affine and color transformations.

Explorations

Non-Semantic Composition from Patches In many human-made collages, the arrangement of patches is determined by their semantic relationship, e.g. a giant child may be depicted climbing atop a skyscraper. The meaning of each part is coherent or interestingly incongruous, and humans can easily construct such scenes. However, a harder task for humans is to compose an image (e.g. a bull or a human face[3]) from semantically different parts (e.g. tree leaves or fruits), as illustrated on Figure 5. CLIP-CLOP easily makes such discoveries and compose patches in a non-semantic way. Figure 5 also shows (in the top row) that fewer patches make more abstract Picasso-inspired collages of a bull, while more patches make more realistic images.

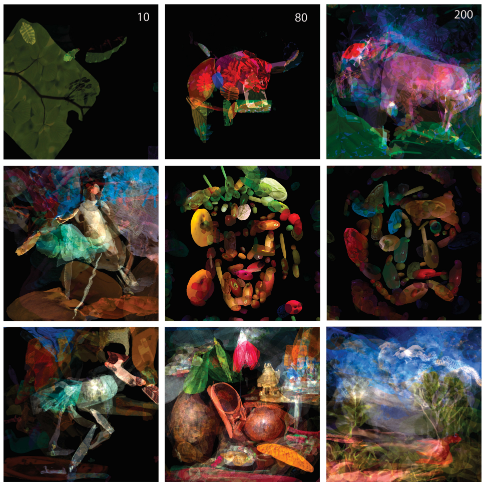


Figure 5: Collages made of different numbers of tree leaves patches (bulls in the top row), as well as Degas-inspired ballet dancers made from animals, faces made of fruit, and still life or landscape made from patches of animals (see Fig. 7).

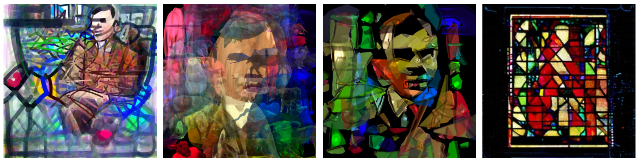


Figure 6: *Alan Turing in stained glass*. a) CLIPDraw with 1000 strokes, b) CLIP-CLOP with 100 animal patches or c) 200 broken plate patches, d) CLIP-guided diffusion.

Patches as Textural Image Constituents Figure 6 illustrates different aesthetics that can be obtained using diverse image generators on the same prompt (*Alan Turing in stained glass*). Without cherry-picking, we compared results on CLIPDraw (Frans, Soros, and Witkowski 2021), Katherine Crowson's CLIP-guided diffusion (Dhariwal and Nichol 2021) using a Colab from late 2021, and CLIP-CLOP on patches consisting of animals or fragments from a broken plate. We noticed that CLIPDraw combines many strokes to create textures, guided diffusion generates complex textures directly from pixels, while collage exploits existing shapes, shadows and textures present on individual patches to achieve the desired stained glass effect.

Generative Collage as a Human-in-the-loop AI Popular applications such as *Wombo Art*[4] that rely on state-of-the-art deep learning for image synthesis, have democratised the use of generative art systems but also removed the human user from most of the image production process, letting them only specify the prompt, and therefore focusing users' input on creative prompt engineering (Liu and Chilton 2022). The user has limited choice in how to visually represent concepts, cannot control the various cultural references and merely acts as a curator of the outputs (Chung 2021). In a departure from commoditized art generation, we propose to give the artist full control over the image patches used for collage, making them curator of the inputs for the algorithm and collaborator with machine creativity. We believe in art as means of human agency, requiring that "automation in creative fields is always supported by the development of humans' creative potential" (Daniele and Song 2019), and thus favour interactive systems over fully automated ones.

Human-in-the-loop systems such as *collabdraw* (Fan, Dinculescu, and Ha 2019) or Drawing Apprentice (Davis, Hsiao, and others 2016) have long been used for AI-guided sketching, and it was found that "AI Steering Tools" for musical composition that let users constrain the generative process "helped users increase their control, creative ownership, and sense of collaboration with the generative ML model" (Louie, Coenen, and others 2020).

In that spirit, we added a simple interactive human-in-the-loop correction of AI-optimized collage. We allow the artist to stop the optimization loop, manually edit one or more patches via a user interface (click on the current collage to select a patch, and adjust its position, rotation, scale, etc. using UI sliders) and then resume the optimization loop.

Conclusion

The remixability of modern media encourages sampling and remixing, hence: collage (Manovich 2005). Collage is yet a little-explored art form for procedural visual art generation. In our work, we introduce a *Collage Generator* and combine it with a popular dual image-and-text encoder like CLIP for AI-based steering. The ability to choose image primitives gives the artist an unprecedented level of control compared to previous CLIP-guided methods and helps to escape, to some extent, the straight-jacket of style imposed by pre-trained neural network generators. Current development work focuses on real-time manipulation of image patches during optimization. We resisted going in the opposite direction: automating the image primitive selection process. We open-source[5] CLIP-CLOP as a creative tool for artists.

[3] A Twitter bot regularly posts collages of human faces, generated by CLIP-CLOP and using patches of animals or human-made waste at: `https://twitter.com/VisPlastica/media`

[4] `https://app.wombo.art`

[5] `https://github.com/deepmind/arnheim`

Acknowledgements

The authors wish to thank Kitty Stacpoole, Christina Lu, Luba Eliott, Max Cant, Jordan Hoffmann, Oriol Vinyals, Ali Eslami, El Morrison, Kory Mathewson, Ian Horton, Mikołaj Bińkowski, Wojciech Stokowiec, Sina Samangooei, JB Alayrac, Yotam Doron, DeepMind colleagues and anonymous reviewers for helpful discussions and suggestions.

Author Contributions

P.M., C.F. and S.O. conceived of the presented idea. P.M., D.B., C.F. and M.M. developed the code. C.F., D.B., P.M. and M.M. generated the artworks. P.M., D.B. and C.F. collected image patches. S.O. supervised this work. All authors contributed to the final manuscript.

Image Patches Used in CLIP-CLOP Collages

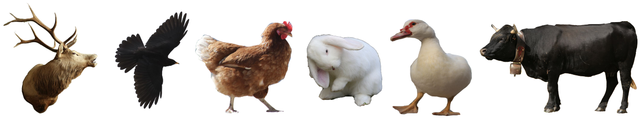


Figure 7: Examples of animal patches. Photos: P. Mirowski. CLIP-CLOP is distributed with copyright-free images (public domain, CC, or photographed or drawn by the authors), see: `https://github.com/deepmind/arnheim`

References

Ades, D. 1976. *Photomontage*. Thames and Hudson London.

Breault, V.; Ouellet, S.; Somers, S.; and Davies, J. 2013. Soilie: A computational model of 2d visual imagination. In *Proceedings of the 11th International Conference on Cognitive Modeling, Ottawa: Carleton University*.

Brock, A.; Donahue, J.; and Simonyan, K. 2018. Large scale gan training for high fidelity natural image synthesis. *arXiv:1809.11096*.

Chung, N. C. 2021. Human in the loop for machine creativity. *arXiv:2110.03569*.

Colton, S. 2008. Creativity versus the perception of creativity in computational systems. In *AAAI spring symposium: creative intelligent systems*, volume 8, 7. Palo Alto, CA.

Crowson, K.; Biderman, S.; Kornis, D.; Stander, D.; Hallahan, E.; Castricato, L.; and Raff, E. 2022. Vqgan-clip: Open domain image generation and editing with natural language guidance. *arXiv preprint arXiv:2204.08583*.

Cunha, J. M.; Martins, P.; and Machado, P. 2018. Emojinating: representing concepts using emoji. In *Workshop Proceedings from ICCBR*.

Daniele, A., and Song, Y.-Z. 2019. Ai+ art= human. In *AAAI/ACM Conference on AI, Ethics, and Society*.

Davis, N.; Hsiao, C.-P.; et al. 2016. Empirically studying participatory sense-making in abstract drawing with a co-creative cognitive agent. In *International Conference on Intelligent User Interfaces*.

Dhariwal, P., and Nichol, A. 2021. Diffusion models beat gans on image synthesis. *arXiv:2105.05233*.

Esser, P.; Rombach, R.; and Ommer, B. 2021. Taming transformers for high-resolution image synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Fan, J. E.; Dinculescu, M.; and Ha, D. 2019. Collabdraw: an environment for collaborative sketching with an artificial agent. In *Creativity and Cognition*.

Fernando, C.; Eslami, S. M. A.; et al. 2021. Generative art using neural visual grammars and dual encoders. *arXiv:2105.00162*.

Frans, K.; Soros, L.; and Witkowski, O. 2021. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. *arXiv:2106.14843*.

Greenberg, C. 1958. The pasted-paper revolution. *Art News*.

Harvey, I. 2009. The microbial genetic algorithm. In *European conference on artificial life*.

Huang, H.; Zhang, L.; and Zhang, H.-C. 2011. Arcimboldo-like collage using internet images. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, 1–8.

Krzeczkowska, A.; El-Hage, J.; Colton, S.; and Clark, S. 2010. Automated collage generation-with intent. In *ICCC*, 36–40.

Liu, V., and Chilton, L. B. 2022. Design guidelines for prompt engineering text-to-image generative models. In *CHI Conference on Human Factors in Computing Systems*, 1–23.

Liu, X.; Gong, C.; et al. 2021. Fusedream: Training-free text-to-image generation with improved clip+ gan space optimization. *arXiv:2112.01573*.

Louie, R.; Coenen, A.; et al. 2020. Novice-ai music co-creation via ai-steering tools for deep generative models. In *CHI Conference on Human Factors in Computing Systems*.

Manovich, L. 2005. Remixing and modularity. *www.manovich.net*.

National Galleries of Scotland. 2019. Cut and paste collage before cubism. https://youtu.be/FKzA5sZBNJw.

Radford, A.; Wook Kim, J.; et al. 2021. Learning transferable visual models from natural language supervision. *arXiv:2103.00020*.

Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2021. High-resolution image synthesis with latent diffusion models.

Tian, Y., and Ha, D. 2021. Modern evolution strategies for creativity: Fitting concrete images and abstract concepts. *arXiv:2109.08857*.

Xiao, P.; Linkola, S. M.; et al. 2015. Vismantic: Meaning-making with images. In *Proceedings of the Sixth International Conference on Computational Creativity*. Brigham Young University.

Zaczek, I., and Actor, M. 2008. *Art: Over 2,500 Works from Cave to Contemporary*. Dorling Kindersley Publishing.

Creative Walk Adversarial Networks: Novel Art Generation with Probabilistic Random Walk Deviation from Style Norms

Divyansh Jha*, **Kai Yi, Ivan Skorokhodov, Mohamed Elhoseiny***
King Abdullah University of Science and Technology (KAUST)
{divyansh.jha, kai.yi, ivan.skorokhodov, mohamed.elhoseiny}@kaust.edu.sa, * denotes Equal contribution.

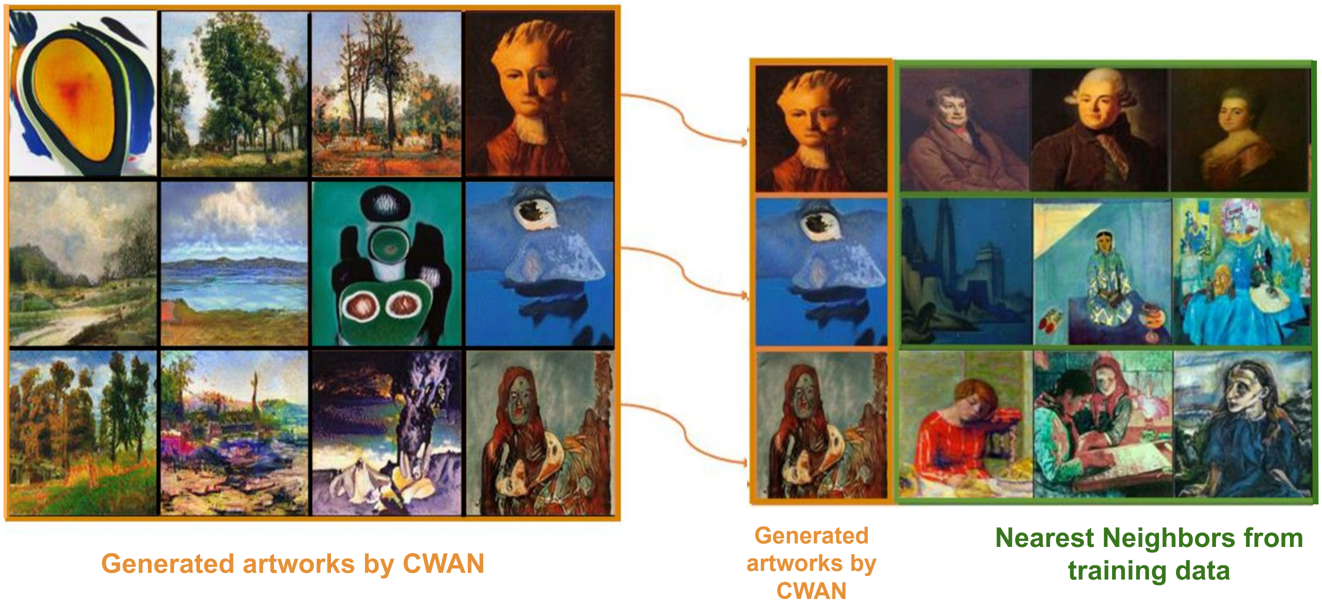

Generated artworks by CWAN
Generated artworks by CWAN
Nearest Neighbors from training data

Figure 1: Art images on left with orange borders are generated using Creative Walk Adversarial Networks. The right part shows the Nearest Neighbors (NN) from the training set on the WikiArt dataset (with green borders), which are different indicating our generations' novelty. Nearest neighbor distance is computed on ResNet-18 space (He et al. 2016).

Abstract

We propose Creative Walk Adversarial Networks (CWAN) for novel art generation. Quality learning representation of unseen art styles is critical to facilitate generation of new meaningful artworks. CWAN learns an improved metric space for generative art by exploring unseen visual spaces with probabilistic random walks. CWAN constructs a dynamic graph that includes the seen art style centers and generated samples in the current minibatch. We then initiate a random walk from each art style center through the generated artworks in the current minibatch. As a deviation signal, we encourage the random walk to eventually land after T steps in a feature representation that is difficult to classify as any of the seen art styles. We investigate the ability of the proposed loss to generate meaningful novel visual art on the WikiArt dataset. Our experimental results and human evaluations demonstrate that CWAN can generate novel art that is significantly more preferable compared to strong state-of-the-art methods, including StyleGAN2 and StyleCAN2. The code is publicly available at: https://vision-cair.github.io/CWAN/

Introduction

Generative models like Generative Adversarial Networks (GANs) (Goodfellow et al. 2014a) and Variational Auto Encoders (VAEs) (Kingma and Welling 2013) are excellent tools for generating images due to their ability to represent high-dimensional probability distributions. However, they are not explicitly trained to go beyond distribution seen during training. Hence, the generations tends to be more emulative than creative. To generate likable novel visual content, GANs' training has been augmented with explicit losses that encourages careful deviation from existing classes, as first demonstrated in Creative Adversarial Networks (CANs) (Elgammal et al. 2017a). These models were shown to have some capability to produce unseen aesthetic art (Elgammal et al. 2017a; Hertzmann 2018; Jha, Chang, and Elhoseiny 2021), fashion (Sbai et al. 2018), design (Nobari, Rashad, and Ahmed 2021), and sculpture (Ge et al. 2019). Producing these creative generations is mainly leveraged by the generative model's improved ability to learn visual representations of novel generations that are distinguishable from seen ones. Similar deviation mechanisms was shown to have generalization benefit, improving performance on the task of unseen class recognition by encouraging discrimination explicitly between seen and unseen generations(Elhoseiny and Elfeki 2019; Elhoseiny, Yi, and Elfeki 2021).

We propose Creative Walk Adversarial Networks (*CWAN*) as a new learning system for generating artworks. We build our method on top of the state-of-the-art GAN architectures, StyleGANs (Karras, Laine, and Aila 2019a; Karras et al. 2020), due to their superior performance as compared to VAEs. We augment StyleGANs with *parameter-free* graph-based loss, dubbed as Creative Walk loss, to improve learning representation of unseen Artworks generatively. We first represent each art style class (e.g., cubism, High renaissance) by its center, representing the mean neural representation of the given Art style. Our Creative Walk loss then starts from the center of each seen art style class and performs a random walk through the generated images for T steps. Then, we encourage the landing representation to be distant and distinguishable from the seen art style centers. In summary, the Creative Walk loss is computed over a similarity graph involving the centers of seen art styles and the generated images/art pieces in the current minibatch. Thus, Creative Walks takes a global view of the data manifold compared to existing deviation losses that are local/per example; e.g., (Sbai et al. 2018; Elgammal et al. 2017a). Our work can be connected to recent advances in semi-supervised learnin, that leverage unlabeled data within the training classes, e.g., (Zhang et al. 2018)(Ayyad et al. 2020)(Ren et al. 2018)(Haeusser, Mordvintsev, and Cremers 2017)(Li et al. 2019). In these methods, unlabeled data are encouraged to attract existing classes. In contrast, our goal is the *opposite*, deviating from seen styles. Also, creative walks operate on generated images instead of provided unlabeled data.

Contribution. We propose Creative Walk Adversarial Networks(CWAN) for novel art generation. CWANs augment state-of-the-art adversarial generative models with a Creative Walk loss that learns an improved metric space for novel art generation. The loss generatively explores unseen art discriminatively against the existing art style classes. The augmented loss is unsupervised on the generative space and can be applied to any GAN architectures; e.g., DCGAN (Radford, Metz, and Chintala 2016), StyleGAN (Karras, Laine, and Aila 2019a), and StyleGAN2 (Karras et al. 2020). We show that Creative Walk Adversarial Networks helps understand unseen visual styles better, improving the generative capability in unseen space of liked art as compared to state-of-the-art baselines including StyleGAN2(Karras et al. 2020) and StyleCAN2(Jha, Chang, and Elhoseiny 2021); see Fig. 1.

Related Work

Generative Models with Deviation Losses. In the context of computational creativity, several approaches have been proposed to produce original items with aesthetic and meaningful characteristics (Machado and Cardoso 2000; Mordvintsev, Olah, and Tyka 2015; DiPaola and Gabora 2009; Tendulkar et al. 2019). Various early studies have made progress on writing pop songs (Briot, Hadjeres, and Pachet 2017), and transferring styles of great painters to other images (Gatys, Ecker, and Bethge 2016; Date, Ganesan, and Oates 2017; Dumoulin et al. 2017; Johnson, Alahi, and Li 2016; Isola et al. 2017) or doodling sketches (Ha and Eck 2018). The creative space of the style transfer images is limited by the content image and the stylizing image, which could be an artistic image by Van Gogh. GANs (Goodfellow et al. 2014a; Radford, Metz, and Chintala 2016; Ha and Eck 2018; Reed et al. 2016; Zhang et al. 2017; Karras et al. 2018; Karras, Laine, and Aila 2019a) have a capability to learn visual distributions and produce images from a latent z vector. However, they are not trained explicitly to produce novel content beyond the training data. More recent work explored an early capability to produce novel art with CAN (Elgammal et al. 2017b), and fashion designs with a holistic CAN (an improved version of CAN) (Sbai et al. 2018), which are based on augmenting DCGAN (Radford, Metz, and Chintala 2016) with a loss encouraging deviation from existing styles. The difference between CAN and holistic-CAN is that the deviation signal is Binary Cross Entropy over individual styles for CAN (Elgammal et al. 2017b) and Multi-Class Cross-Entropy (MCE) loss overall styles in Holistic-CAN (Sbai et al. 2018). (Jha, Chang, and Elhoseiny 2021) recently proposed *StyleCAN* model, which augments the Holistic CAN loss on StyleGANs, showing an improved performance compared to StyleGANs in art generation.

In contrast to these deviation losses, our Creative Walk loss is global. It establishes dynamic messages between generations produced in every mini-batch iteration and seen visual spaces. These generations deviate from style norms represented by the centers of the seen art style classes. In our experiments, we added the proposed loss to StyleGAN1 and StyleGAN2 architectures to produce novel visual artworks, showing superior likeability compared to existing losses.

Creative Walk Adversarial Networks

We start this section by the formulation of our *Creative Walk* loss. We will show later in this section how state-of-the-art deep-GAN models can be integrated to encourage novel visual generations. We denote the generator as $G(z)$ and its corresponding parameters as θ_G. As in (Goodfellow et al. 2014b; Karras, Laine, and Aila 2019a), the random vector $z \in \mathbb{R}^Z$ sampled from a Gaussian distribution $p_z = \mathcal{N}(0,1)$ to generate an image. Hence, $G(z)$ is an generated image from the noise vector z. We denote the discriminator as D and its corresponding parameters as θ_D. The discriminator is trained with two objectives: (1) predicting real images from the training images and fake for generated ones. (2) identify the art style class of the input artwork. The discriminator then has two classification heads. The first head is for binary real/fake classification; $\{0,1\}$ classifier. The second head is a K-way classifier over the seen art style classes, where K is the number of style classes in the training dataset. We denote the real/fake probability produced by D for an input image as $D^r(\cdot)$, and the classification score of a seen style class $k \in \mathcal{S}$ given the image as $D^c(\cdot)$, where $\mathcal{S}$ is the set of seen art styles.

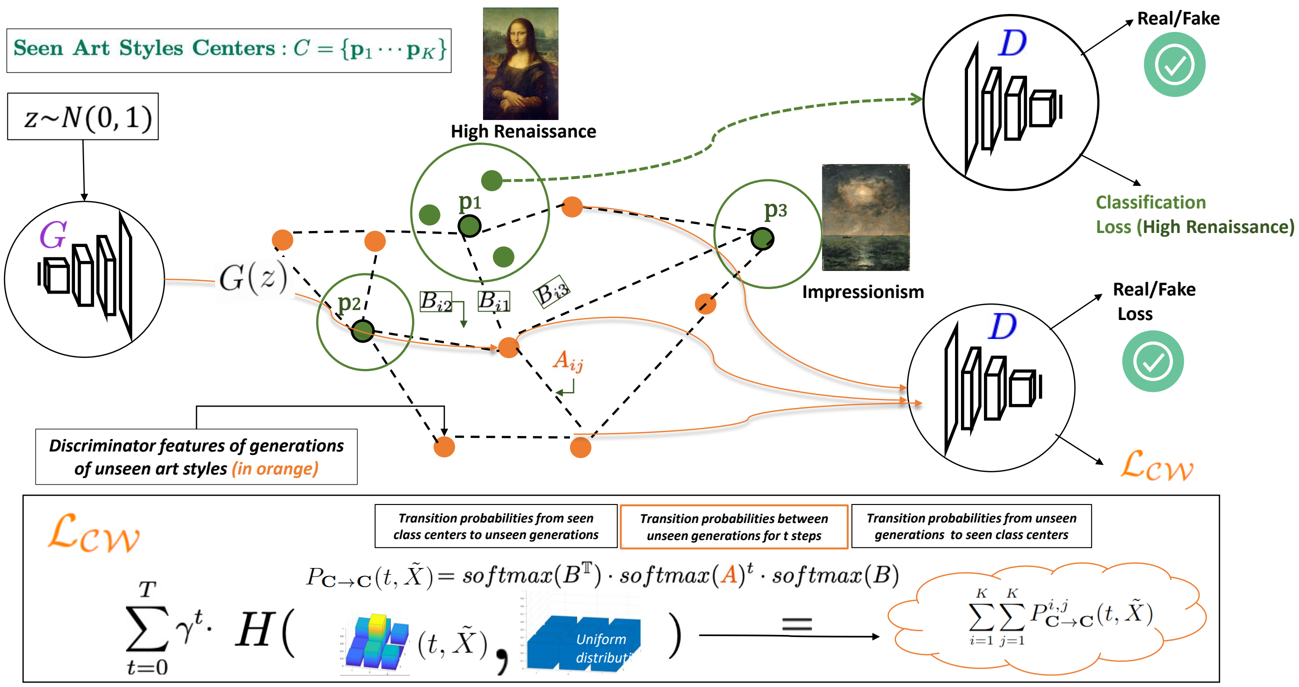

Seen Art Styles Centers : $C = \{\mathbf{p}_1 \cdots \mathbf{p}_K\}$
$z \sim N(0,1)$
G
$G(z)$
High Renaissance
p1
p2
p3
Impressionism
B_{i2}
B_{i1}
B_{i3}
A_{ij}
D
Real/Fake
Classification Loss (High Renaissance)
D
Real/Fake Loss
$\mathcal{L}_{CW}$
Discriminator features of generations of unseen art styles (in orange)
$\mathcal{L}_{CW}$
Transition probabilities from seen class centers to unseen generations
Transition probabilities between unseen generations for t steps
Transition probabilities from unseen generations to seen class centers
$P_{\mathbf{C}\to\mathbf{C}}(t,\tilde{X}) = softmax(B^{\mathbb{T}}) \cdot softmax(A)^t \cdot softmax(B)$
$\sum_{t=0}^{T} \gamma^t \cdot H(\quad (t,\tilde{X}), \quad)$
Uniform distributi
$\sum_{i=1}^{K}\sum_{j=1}^{K} P^{i,j}_{\mathbf{C}\to\mathbf{C}}(t,\tilde{X})$

Figure 2: Creative Walk loss starts from each seen style class center (i.e., $\mathbf{p}_i$). It then performs a random walk through generated examples of hallucinated unseen classes using $G(z)$ for T steps. The landing probability distribution of the random walk is encouraged to be uniform over the seen classes. For careful deviation from seen classes, the generated images are encouraged to be classified as real by the Discriminator D. H indicates relative entropy; see Eq. 4 for detailed definition.

Creative Walk Loss

We denote the seen class centers, or prototypes[1], that we aim to deviate from as $C = \{\mathbf{p}_1 \cdots \mathbf{p}_K\}$, where $\mathbf{p}_i$ represents center of seen class/style i and K is the number of seen art styles that we aim to deviate from. We compute $C = \{\mathbf{p}_1 \cdots \mathbf{p}_K\}$ by sampling a small episodic memory of size m for every class and computing $\mathbf{p}_i$ from the discriminator representation. Concretely, we randomly sample $m = 10$ examples per class once and compute at each iteration its mean discriminator features, computed as activations from the last layer of the Discriminator D followed by scaled L2 normalization $L2(\mathbf{v}, \beta) = \beta \frac{\mathbf{v}}{\|\mathbf{v}\|}$, $\beta = 3$.

With the generator $G(\cdot)$, we sample generated images $\tilde{X}$ of size $\tilde{N}$ that we aim them to deviate from the seen art styles. $\tilde{X}$ is then embedded to the same feature space as style centers with the discriminator. Let $B \in \mathbb{R}^{\tilde{N}\times K}$ be the similarity matrix between the features of the generations ($\tilde{X}$) and the seen class centers (C). Similarly, let $A \in \mathbb{R}^{\tilde{N}\times\tilde{N}}$ be the similarity matrix between the generated images. In particular, we use the negative Euclidean distances between the embeddings as a similarity measure as follows:

$$B_{ij} = -\|\tilde{x}_i - \mathbf{p}_j\|^2, \quad A_{i,j} = -\|\tilde{x}_i - \tilde{x}_j\|^2 \qquad (1)$$

where $\tilde{x}_i$ and $\tilde{x}_j$ are i^{th} and j^{th} features in the set $\tilde{X}$; see Fig. 2. To avoid self-cycle, The diagonal entries $A_{i,i}$ are set to a small number ϵ.

Hence, we defined three transition probability matrices:

$$P_{\mathbf{C}\to\tilde{X}} = \sigma(B^{\mathbb{T}}), \;\; P_{\tilde{X}\to\mathbf{C}} = \sigma(B), \;\; P_{\tilde{X}\to\tilde{X}} = \sigma(A) \qquad (2)$$

where σ is the softmax operator is applied over each row of the input matrix, $P_{\mathbf{C}\to\tilde{X}}$ and $P_{\tilde{X}\to\mathbf{C}}$ are the transition probability matrices from each seen class over the $\tilde{N}$ generated images and vice-versa respectively. $P_{\tilde{X}\to\tilde{X}}$ is the transition probability matrix from each generated image over other generated images. We hence define our generative random walker probability matrix as:

$$P_{\mathbf{C}\to\mathbf{C}}(t,\tilde{X}) = P_{\mathbf{C}\to\tilde{X}} \cdot (P_{\tilde{X}\to\tilde{X}})^t \cdot P_{\tilde{X}\to\mathbf{C}} \qquad (3)$$

where $P^{i,j}_{\mathbf{C}\to\mathbf{C}}(t,\tilde{X})$ denotes the probability of ending a random walk of a length t at a seen class j given that we have started at seen class i; t denotes the number of steps taken between the generated points, before stepping back to land on a seen art style.

Creative Walk Loss. Our random walk loss aims at boosting the deviation of unseen visual spaces from seen art style classes. Hence, we define our loss by encouraging each row in $P_{\mathbf{C}\to\mathbf{C}}(t)$ to be hard to classify to seen classes as follows

$$\begin{aligned} L_{CW} = &-\sum_{t=0}^{T} \gamma^t \cdot \sum_{i=1}^{K}\sum_{j=1}^{K} U_c(j) log(P^{i,j}_{\mathbf{C}\to\mathbf{C}}(t,\tilde{X})) \\ &-\sum_{j=1}^{N_u} U_x(j) log(P_v(j)) \end{aligned} \qquad (4)$$

where the first term minimizes cross entropy loss between every row in $P_{\mathbf{C}\to\mathbf{C}}(t,\tilde{X}) \forall t = 1 \to T$ and uniform distribution over seen classes $U_c(j) = \frac{1}{K^s}, \forall j = 1 \cdots K^s$, where

[1] we refer alternatively between prototypes and centers

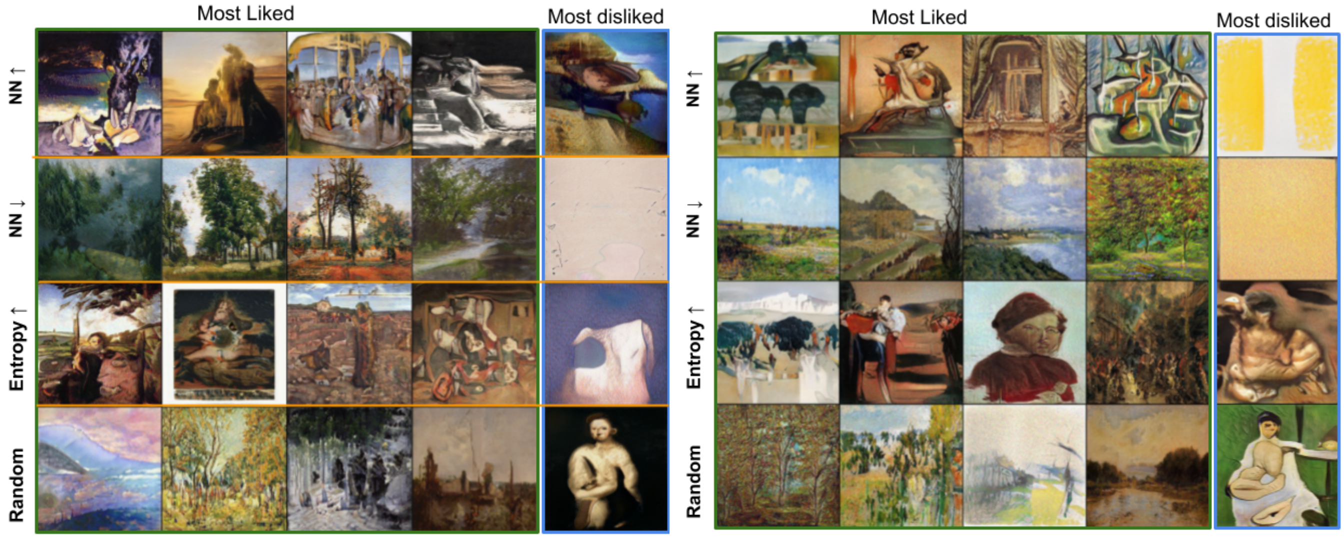


Figure 3: Most liked and disliked art generated using StyleGAN1 + CWAN(left) and StyleGAN2 + CWAN(right) architectures.

T is a hyperparameter and γ is exponential decay set to 0.7 in our experiments. In the second term, we maximize the probability of all the generations $\tilde{x}_i \in \tilde{X}$ to be equality visited by the random walk; see Fig. 2. This term is called the “visit loss” and was proposed in (Haeusser, Mordvintsev, and Cremers 2017) to encourage random walker to visit a large set of unlabeled points. We compute the overall probability that each generated point would be visited by any of the seen class $P_v = \frac{1}{K} \sum_{i=0}^{K} P^i_{\mathbf{C} \to \tilde{X}}$, where $P^i_{\mathbf{C} \to \tilde{X}}$ represents the i^{th} row of the $P^{\mathbf{C} \to \tilde{X}}$ matrix. The visit loss is then defined as the cross-entropy between P_v and the uniform distribution $U_x(j) = \frac{1}{\tilde{N}}, \forall j = 1 \cdots \tilde{N}$. Hence, visit loss encourages to visit as many examples as possible from $\tilde{X}$ and hence improves learning representation.

Note that, if we replace U_c by an identity matrix to encourage landing to the starting seen class, the loss becomes an attraction signal similar to (Haeusser, Mordvintsev, and Cremers 2017), which defines its conceptual difference to the Creative Walk loss. We integrated our loss with StyleGAN1 (Karras, Laine, and Aila 2019a) and StyleGAN2 (Karras et al. 2020) by simply adding L_{GRW} in Eq. 4 to the generator loss. The generator and discriminator losses are defined as follows

$$\mathcal{L}_G = \mathcal{L}_{G\text{ GAN}} + \lambda \mathcal{L}_{CW} \tag{5}$$

$$\mathcal{L}_D = \mathcal{L}_{D\text{ GAN}} + \lambda \mathcal{L}_{\text{style\_classification}} \tag{6}$$

where $\mathcal{L}_{G\text{ GAN}}$ and $\mathcal{L}_{D\text{ GAN}}$ are the default generator and discriminator loss, used in the adopted GAN architecture (e.g., DCGAN, StyleGAN1. StyleGAN2). Similar to (Elgammal et al. 2017a; Sbai et al. 2018), we add art style classification loss, $\mathcal{L}_{\text{style\_classification}}$, on real art images to $\mathcal{L}_D$.

Experiments

Dataset. We performed our experiments on the WikiArt datasets (WikiArt 2015), which contains approximately 81k art works from 27 different art styles and over 1k artists.

Nomenclature. Our models are referred as CWAN-T(value), where CWAN means Creative Walk Adversarial Network, with Creative Walk loss of T time steps. We name our models according to this convention throughout this section. We perform human subject experiments to evaluate generated art. We set value of the loss coefficient λ as 10 in all our experiments. We divide the generations from these models into four groups, each containing 100 images; see examples in Fig. 3.

- **NN↑.** Images with high nearest neighbor (NN) distance from the training dataset.
- **NN↓.** Images with low nearest neighbor (NN) distance from the training dataset.
- **Entropy ↑.** Images with high entropy of the probabilities from a style classifier trained on WikiArt dataset.
- **Random (R).** A set of random images.

For example, we denote generations using CWAN with T=10, and NN↑ group as CWAN-T10_NN↑. Fig. 3 shows top liked/disliked paintings according to human evaluation on StyleGAN1 and StyleGAN2 with our Creative Walk loss.

Baselines. We performed comparisons with two baselines, i.e., (1) the vanilla GAN for the chosen architecture, and (2) adding Holistic-CAN loss (Sbai et al. 2018) (an improved version of CAN (Elgammal et al. 2017b)). For simplicity, we refer the Holistic-CAN as CAN. We also compared to StyleCAN(Jha, Chang, and Elhoseiny 2021) model, an adaptation of the holistic CAN loss on the state-of-the-art StyleGAN (Karras, Laine, and Aila 2019b) and StyleGAN2 (Karras et al. 2020) architectures.

Human Evaluation. We performed our human subject MTurk experiments based on StyleGAN1 (Karras, Laine, and Aila 2019b) & StyleGAN2 (Karras et al. 2020) architecture's vanilla GAN, CAN, and CWAN variants. We conducted three types of experiments; see Fig. 5.

1. **Likeability Experiment:** Following(Elgammal et al. 2017a), we performed the likeability experiments on

Table 1: Human experiments on generated art from vanilla GAN, and CAN, and CWAN. CWAN obtained the highest mean likeability in all the groups. Here Q1 is asking for a likeability score and Q2 is asking whether the art work is created by a computer/human. See Likeability Experiment for more details. More people believed the generated art to be real for the artwork generated from model trained using the Creative Walk loss.

		Likeability Mean					Turing Test
Loss	Architecture	Q1-mean(std)	NN ↑	NN ↓	Entropy ↑	Random	Q2(% Artist)
CAN (Elgammal et al. 2017b)	DCGAN	3.20(1.50)	-	-	-	-	53
GAN (Vanilla) (Karras, Laine, and Aila 2019a)	StyleGAN	3.12(0.58)	3.07	3.36	3.00	3.06	55.33
CAN (Jha, Chang, and Elhoseiny 2021)	StyleGAN	3.20(0.62)	3.01	3.61	3.05	3.11	56.55
CWAN-T3 (Ours)	StyleGAN	**3.29(0.59)**	3.05	3.58	3.13	**3.38**	54.08
CWAN-T10 (Ours)	StyleGAN	**3.29(0.63)**	**3.15**	**3.67**	**3.15**	3.17	**58.63**
GAN (Vanilla) (Karras et al. 2020)	StyleGAN2	3.02(1.15)	2.89	3.30	2.79	3.09	54.01
CAN (Jha, Chang, and Elhoseiny 2021)	StyleGAN2	3.23(1.16)	3.27	3.34	3.11	3.21	57.9
CWAN-T3 (Ours)	StyleGAN2	**3.40(1.1)**	**3.30**	**3.61**	**3.33**	**3.35**	**64.0**

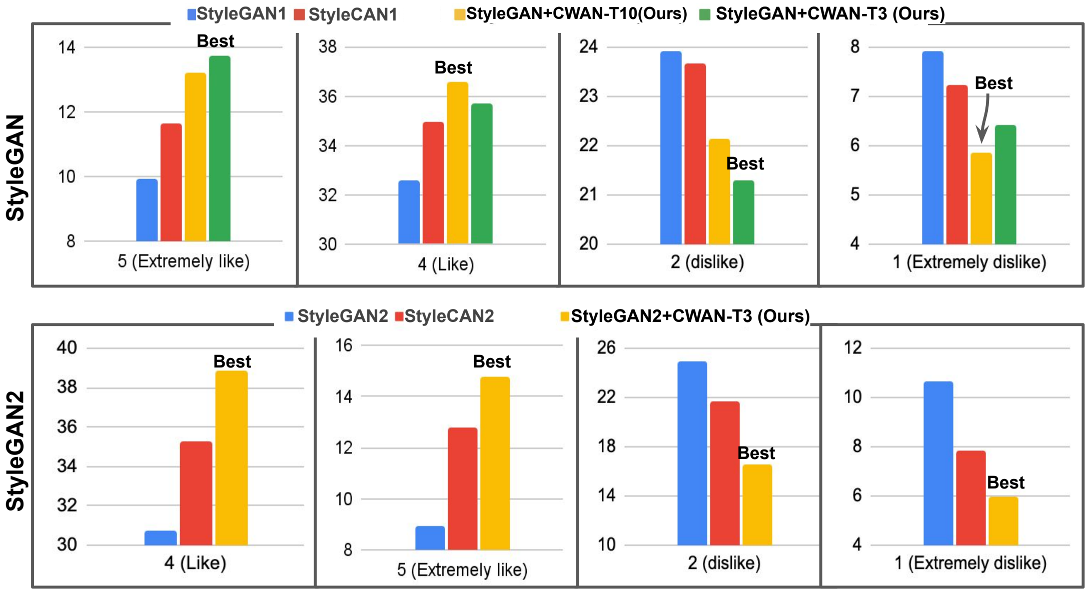


Figure 4: Percentage of each rating from human subject experiments on generated images. Compared to CAN, images generated using CWAN are rated (5,4) by a significantly larger share of people, and are rated (1,2) by fewer people.

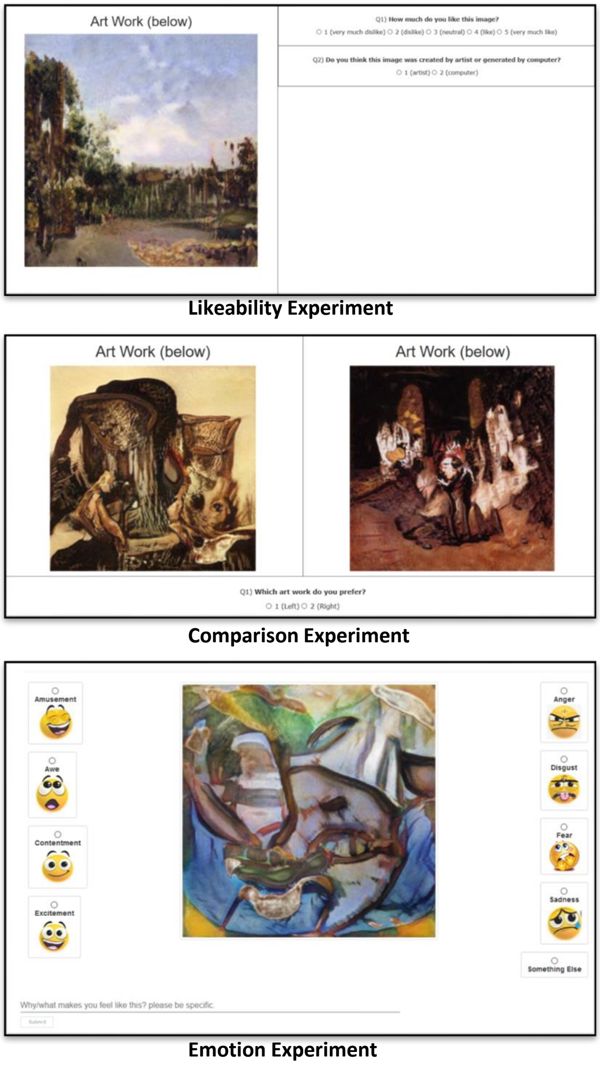
Amazon Mechanical Turk by asking the surveyors the following questions.

(a) **Q1.** How much do you like this image? 1-5 rating; 5 is best rating.

(b) **Q2.** Do you think this image was created by artist or generated by computer? (yes/no)

The user interface of this experiment is shown in Figure 5 (top). We divide the generations into four groups described in nomenclature. We collect five responses for each art piece (400 images), totaling 2000 responses per model by 341 unique workers. Table 1 summarizes the likeability of CWAN generated artworks in comparison to vanilla GAN and StyleCAN variants (Jha, Chang, and Elhoseiny 2021). We find that images generated from our model is more likeable in all the groups described earlier. Figure 4 shows how our paintings are given higher ratings by more share of participants and lower ratings by less participants. We found that artworks from the trained StyleGAN1 and StyleGAN2 with our Creative Walk loss were more likeable and more people believed them to be real art, as shown in Table 1. For StyleGAN1, adding the Creative Walk loss resulted in 38% and 18% more people giving a full rating of 5 over vanilla StyleGAN1 and StyleGAN1 + CAN (StyleCAN1) loss, respectively, see Fig. 4. For StyleGAN2, these improvements are 65% and 15%. Table 2 shows that images generated by CWAN on StyleGAN1 and StyleGAN2 architectures have better ranks when combined with sets from other baselines.

2. **Comparison Experiment:** We performed experiments where given an artwork from a model trained with our Creative Walk loss vs an artwork with CAN loss, we ask people, which one they prefer. The pairing of the im-



Q1) **How much do you like this image?**

○ 1 (very much dislike) ○ 2 (dislike) ○ 3 (neutral) ○ 4 (like) ○ 5 (very much like)

Q2) **Do you think this image was created by artist or generated by computer?**

○ 1 (artist) ○ 2 (computer)

Likeability Experiment

Q1) **Which art work do you prefer?**

○ 1 (Left) ○ 2 (Right)

Comparison Experiment

Why/what makes you feel like this? please be specific.

Emotion Experiment

Figure 5: User interfaces of the likeability experiment(top), comparison experiment(middle) and emotion experiment(bottom).

Table 2: Normalized mean ranking (lower the better) calculated from the likeability experiment. We take the mean rating of each artwork on both CAN and CWAN losses. We then stack, sort, normalize them to compute the normalized mean rank. The numbers are corresponding normalized ranks from the models in the row above them.

	Normalized Mean Ranks		
	CAN/**CWAN-T10**	CAN/**CWAN-T3**	CAN/**CWAN-T10**/**CWAN-T3**
StyleGAN1	0.53/**0.47**	0.53/**0.47**	0.52/**0.48**/**0.50**
	CAN/**CWAN-T3**	GAN/**CWAN-T3**	CAN/GAN/**CWAN-T3**
StyleGAN2	0.54/**0.46**	0.59/**0.41**	0.49/0.59/**0.42**

ages was done on the basis of nearest neighbour. So, for each image generated from a StyleGAN model trained on Creative Walk loss, we found the nearest neighbour from images of model trained on CAN loss. Several qualitative results from these experiments are shown in Figure 6. The nearest neighbour was computed based on features that were extracted from a pretrained ResNet-18 (He et al. 2016). This is to make sure that the images we give out for comparison looks similiar as possible. We took random pairs of images from generations from StyleGAN model trained with CAN and CWAN; see the user interface for this experiment in Figure 5 (middle). The results for this experiment on StyleGAN 1 and 2 model on CWAN and CAN losses are summarized in Table 3. We collected 5 responses each for 600 pairs of artworks by 300 unique workers. Table 3 shows that CWAN loss is significantly more preferred compared to art work from CAN losses.

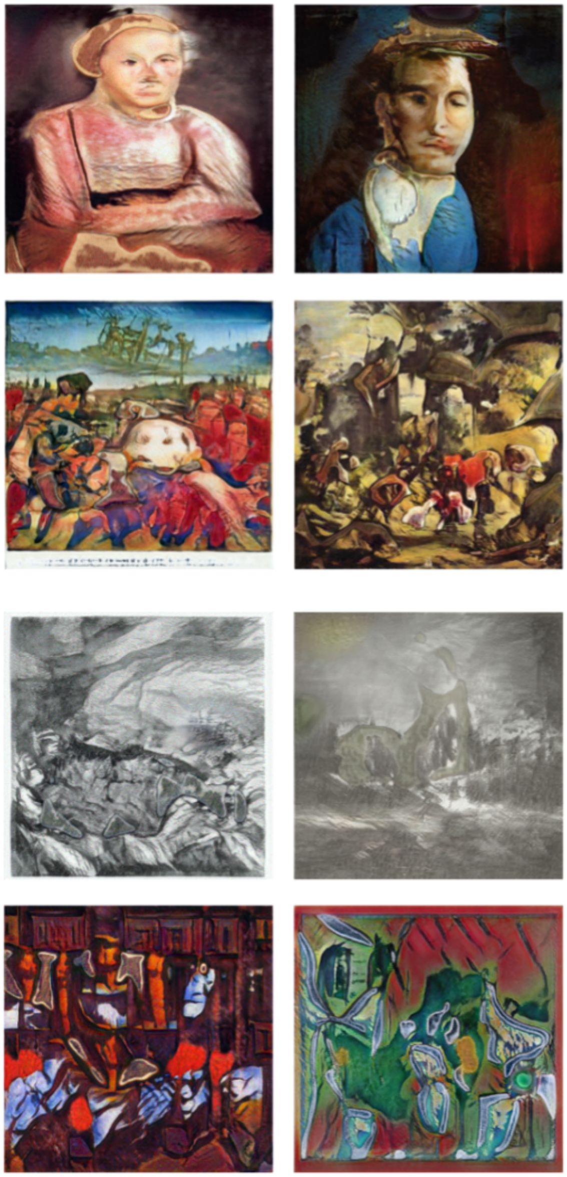


Figure 6: Figure shows CWAN (left) preferred more than CAN (right) for each pair of columns (random selection).

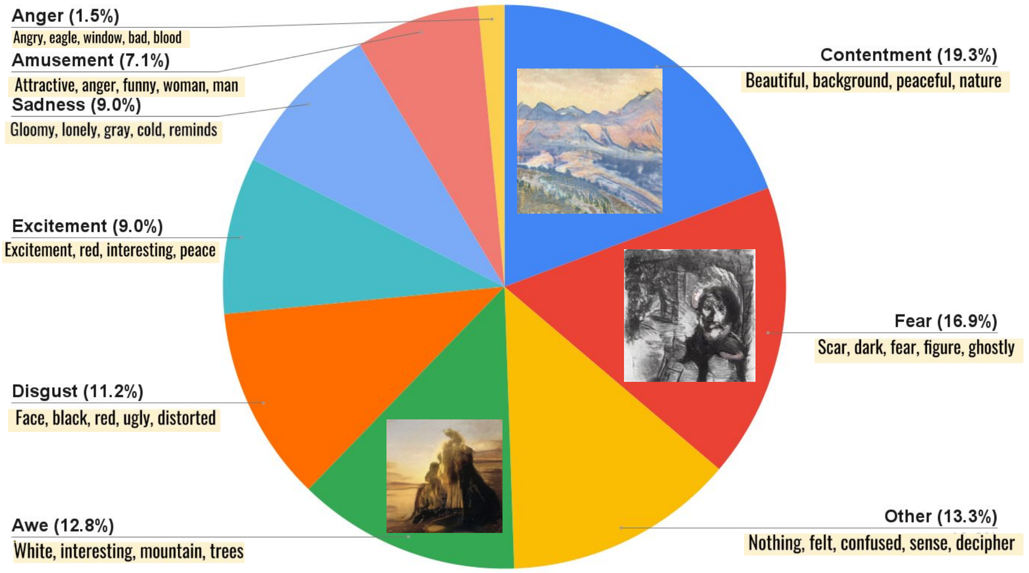


Figure 7: Distribution of emotional responses for generated art from StyleGAN1 + CWAN. Example image for fear, awe, and contentment is shown. The box beneath shows the most frequent words used by evaluators to describe their feeling. These responses were collected from a survey on Amazon Mechanical Turk.

3. **Emotion Human Subject Experiments:** To measure the emotional influence of AI generated art on the participants similar to (Jha, Chang, and Elhoseiny 2021), we asked participants to record their constructed emotion when exposed to a generated artwork. Following (Machajdik and Hanbury 2010; Achlioptas et al. 2021; Mohamed et al. 2022), we allowed these options of emotion categories 1) Amusement 2) Awe 3) Contentment 4) Excitement 5) Anger 6) Disgust 7) Fear 8) Sadness and 9) Something Else (“Other” in Fig 7). People were also asked to describe why they feel that particular emotion in text, so that survey participants chose the emotion after properly looking at the art work; see the user interface Figure 5 (bottom). We collected 5 responses each for a set of 400 generated artworks from 250 unique workers. Despite the model being trained unconditionally, it was able to produce generations that constructed diverse feelings in the viewer. Fig. 7 shows the distribution over the opted emotions, which are diverse but mostly positive. However, some generations construct negative emotions like fear. Fig. 7 also shows the most frequent words for each emotion after removing stop words. Notable positive words include “funny”, “beautiful”, “attractive”, and negative words include “dark”, “ghostly” which are associated with feelings like fear and disgust. Compared to the emotion experiments on Real Art and StyleCAN reported in (Jha, Chang, and Elhoseiny 2021), emotional responses to StyleGAN +CWAN art are more entropic (diverse).

Table 3: Evaluator preference percentage for generated images for both CWAN and CAN loss on the StyleGAN architectures. We split the preferred images into two groups based on their NN distance, and then the preference percentage is calculated for these groups.

	Architecture	Low NN distance split	High NN distance split
CAN	StyleGAN1	0.46	0.48
CWAN-T10	StyleGAN1	**0.54**	**0.52**
CAN	StyleGAN2	0.46	0.43
CWAN-T3	StyleGAN2	**0.54**	**0.56**

Emotional Descriptions by people. In Fig. 9, we can see a sample of the emotional descriptions that we collected on the art generated by CWAN in the emotion human subject experiment. One of the interesting descriptions we collect by a survey participant where they describe an artwork with a old looking female as ”Zombie grandma”. Another survey participant describes a artwork generated as ”super relaxing” because of the sunset like colors in the artwork. More examples are shown in Fig. 9

Wundt Curve Analysis. Wundt curve (Packard 1975; Wundt 1874) illustrates Collin Martinale’s “The principle of least efforts”, a theory that explains human behavior towards creativity in artworks (Martindale 1990). The curve shows that as the originality/novelty of the work increases, people like the work. After a certain threshold, people start disliking it due to the difficulty of understanding, which leads to a lack of appreciation. We approximate Wundt curve by fitting a degree 3 polynomial on a scatter plot of normalized likeability vs. mean NN distance (novelty measure). Generations are more likable if the deviation from existing art is moderate but not too much; see Fig. 8. We observe that likeability responses to image sets with higher NN distance (i.e., Random (R) and NN↑) are generally lower compared to NN↓. Compared to CAN and GAN, CWAN achieves on

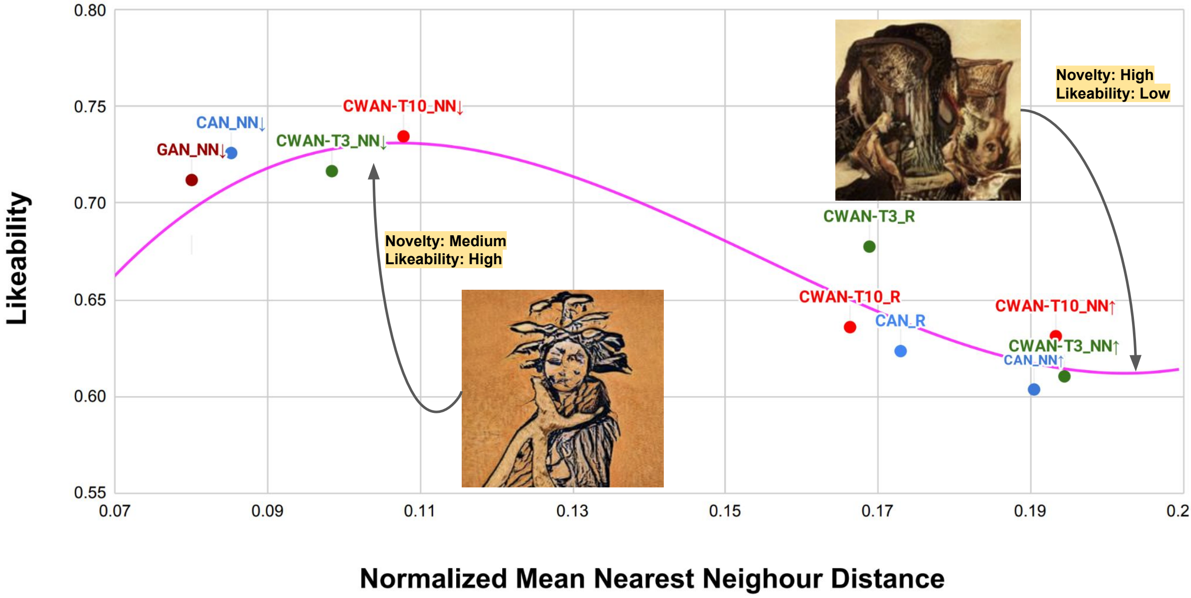
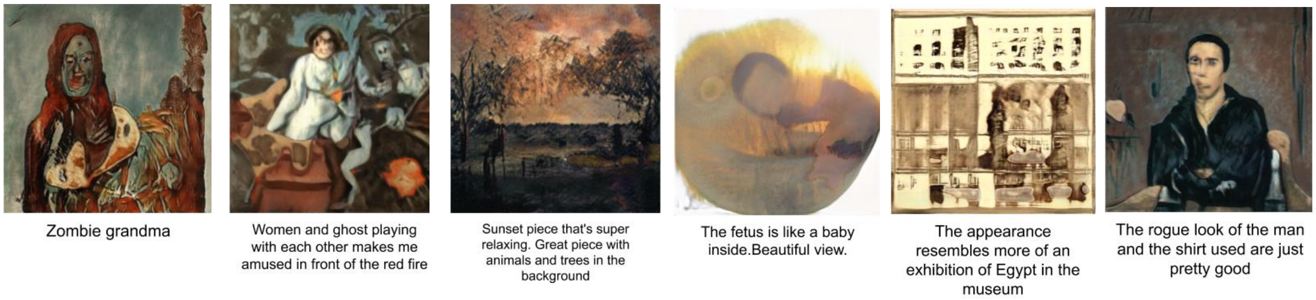


Figure 8: Empirical approximation of Wundt Curve (Packard 1975; Wundt 1874). The color of the data point represents a specific model and its label specifies the group named according to nomenclature. Art from the NN ↑ group has lower likeability than the NN ↓ group. Examples of a high and low likeability artwork and its novelty are shown. The NN distance is computed from features of resnet-18 and are normalized by scaling down by 20 (to be < 1). We select 20 because it was around the higher NN distances we observe in our generations


Zombie grandma

Women and ghost playing with each other makes me amused in front of the red fire

Sunset piece that's super relaxing. Great piece with animals and trees in the background

The fetus is like a baby inside.Beautiful view.

The appearance resembles more of an exhibition of Egypt in the museum

The rogue look of the man and the shirt used are just pretty good

Figure 9: Descriptions given by people when asked to describe the why they felt a particular emotion while looking at artworks generated by CWAN (our method)

balance novel images that are more preferred.

Key Observations

In the experiments and the analysis conducted above, we noted the following key observations.

1. *The creative walk loss used in CWAN has performed better than CAN on two SOTA base architectures i.e. StyleGAN1 and StyleGAN2.*
2. *From Table 1 we find that the artworks generated by our proposed CWAN model are more likeable than those artworks by CAN in all the evaluation groups.*
3. *From Fig. 3 we see that artworks by CWAN have a significantly higher percentage of people giving a rating of 5 and least amount for people giving a rating of 1.*
4. *In Fig. 8, we approximated the Wundt Curve from artworks generated from CWAN.*
5. *The generated artworks were able to construct meaningful and diverse emotional experiences for our human participants as shown in Figures 7 and 9*

Conclusion

We propose Creative Walk Adversarial Networks. Augmenting Generative Adversarial Networks with a creative random walk loss. Through our experiments and analysis, we showed that CWAN improves generative models' capability to better represent the unseen artistic space and generate preferable novel artworks. We think the improvement is due to our learning mechanism's global nature, which operates at the minibatch level producing generations that are message-passing to each other to facilitate better deviation of generated artworks from seen art style classes.

Author Contributions

Mohamed Elhoseiny came up with the idea and Divyansh Jha was involved more in the implementation. Both Divyansh and Mohamed invested time in writing different sections of the paper. Where Mohamed helped in the theoretical aspects, Divyansh worked on experiments, figures and observations. Mohamed also played a key role in validation of results and observations. The other two authors Kai and Ivan helped us in paper writing for some duration of the life of the project.

Acknowledgements

This work is funded KAUST BAS/1/1685-01-01. We thank the IBEX computing cluster team at KAUST for their continual support in providing us resources and helping with technical challenges. We also wish to emphasize our gratitude to all the hard working Amazon Mechanical Turkers without whom this work would not be possible.

References

[Achlioptas et al. 2021] Achlioptas, P.; Ovsjanikov, M.; Haydarov, K.; Elhoseiny, M.; and Guibas, L. J. 2021. Artemis: Affective language for visual art. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11569–11579.

[Ayyad et al. 2020] Ayyad, A.; Navab, N.; Elhoseiny, M.; and Albarqouni, S. 2020. Semi-supervised few-shot learning with prototypical random walks.

[Briot, Hadjeres, and Pachet 2017] Briot, J.-P.; Hadjeres, G.; and Pachet, F. 2017. Deep learning techniques for music generation-a survey. *arXiv:1709.01620*.

[Date, Ganesan, and Oates 2017] Date, P.; Ganesan, A.; and Oates, T. 2017. Fashioning with networks: Neural style transfer to design clothes. In *KDD ML4Fashion workshop*.

[DiPaola and Gabora 2009] DiPaola, S., and Gabora, L. 2009. Incorporating characteristics of human creativity into an evolutionary art algorithm. *Genetic Programming and Evolvable Machines* 10(2):97–110.

[Dumoulin et al. 2017] Dumoulin, V.; Shlens, J.; Kudlur, M.; Behboodi, A.; Lemic, F.; Wolisz, A.; Molinaro, M.; Hirche, C.; Hayashi, M.; Bagan, E.; et al. 2017. A learned representation for artistic style. *ICLR*.

[Elgammal et al. 2017a] Elgammal, A.; Liu, B.; Elhoseiny, M.; and Mazzone, M. 2017a. Can: Creative adversarial networks, generating" art" by learning about styles and deviating from style norms. *arXiv preprint arXiv:1706.07068*.

[Elgammal et al. 2017b] Elgammal, A.; Liu, B.; Elhoseiny, M.; and Mazzone, M. 2017b. Can: Creative adversarial networks, generating" art" by learning about styles and deviating from style norms. In *International Conference on Computational Creativity*.

[Elhoseiny and Elfeki 2019] Elhoseiny, M., and Elfeki, M. 2019. Creativity inspired zero-shot learning. In *Proceedings of the IEEE International Conference on Computer Vision*, 5784–5793.

[Elhoseiny, Yi, and Elfeki 2021] Elhoseiny, M.; Yi, K.; and Elfeki, M. 2021. Cizsl++: Creativity inspired generative zero-shot learning. *arXiv preprint arXiv:2101.00173*.

[Gatys, Ecker, and Bethge 2016] Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2016. Image style transfer using convolutional neural networks. In *CVPR*.

[Ge et al. 2019] Ge, S.; Dill, A.; Kang, E.; Li, C.-L.; Zhang, L.; Zaheer, M.; and Poczos, B. 2019. Developing creative ai to generate sculptural objects. *arXiv preprint arXiv:1908.07587*.

[Goodfellow et al. 2014a] Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014a. Generative adversarial nets. In *NIPS*, 2672–2680.

[Goodfellow et al. 2014b] Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014b. Generative adversarial networks. In *NIPS*.

[Ha and Eck 2018] Ha, D., and Eck, D. 2018. A neural representation of sketch drawings. *ICLR*.

[Haeusser, Mordvintsev, and Cremers 2017] Haeusser, P.; Mordvintsev, A.; and Cremers, D. 2017. Learning by association — a versatile semi-supervised training method for neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 626–635.

[He et al. 2016] He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

[Hertzmann 2018] Hertzmann, A. 2018. Can computers create art? In *Arts*, volume 7, 18. Multidisciplinary Digital Publishing Institute.

[Isola et al. 2017] Isola, P.; Zhu, J.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. *CVPR*.

[Jha, Chang, and Elhoseiny 2021] Jha, D.; Chang, H. H.; and Elhoseiny, M. 2021. Wölfflin's affective generative analysis of visual art. *The International Conference on Computational Creativity (ICCC)*.

[Johnson, Alahi, and Li 2016] Johnson, J.; Alahi, A.; and Li, F. 2016. Perceptual losses for real-time style transfer and super-resolution. *ECCV*.

[Karras et al. 2018] Karras, T.; Aila, T.; Laine, S.; and Lehtinen, J. 2018. Progressive growing of GANs for improved quality, stability, and variation. *ICLR*.

[Karras et al. 2020] Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; and Aila, T. 2020. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8110–8119.

[Karras, Laine, and Aila 2019a] Karras, T.; Laine, S.; and Aila, T. 2019a. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4401–4410.

[Karras, Laine, and Aila 2019b] Karras, T.; Laine, S.; and Aila, T. 2019b. A style-based generator architecture for generative adversarial networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).*

[Kingma and Welling 2013] Kingma, D. P., and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

[Li et al. 2019] Li, X.; Sun, Q.; Liu, Y.; Zhou, Q.; Zheng, S.; Chua, T.-S.; and Schiele, B. 2019. Learning to self-train for semi-supervised few-shot classification. In *Advances in Neural Information Processing Systems*, 10276–10286.

[Machado and Cardoso 2000] Machado, P., and Cardoso, A. 2000. Nevar–the assessment of an evolutionary art tool. In *Proc. of the AISB00 Symposium on Creative & Cultural Aspects and Applications of AI & Cognitive Science*, volume 456.

[Machajdik and Hanbury 2010] Machajdik, J., and Hanbury, A. 2010. Affective image classification using features inspired by psychology and art theory. In *Proceedings of the 18th ACM international conference on Multimedia*, 83–92.

[Martindale 1990] Martindale, C. 1990. *The clockwork muse: The predictability of artistic change.* Basic Books.

[Mohamed et al. 2022] Mohamed, Y.; Khan, F. F.; Haydarov, K.; and Elhoseiny, M. 2022. It is okay to not be okay: Overcoming emotional bias in affective image captioning by contrastive data collection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[Mordvintsev, Olah, and Tyka 2015] Mordvintsev, A.; Olah, C.; and Tyka, M. 2015. Inceptionism: Going deeper into neural networks. *Google Research Blog. Retrieved June*.

[Nobari, Rashad, and Ahmed 2021] Nobari, A. H.; Rashad, M. F.; and Ahmed, F. 2021. Creativegan: editing generative adversarial networks for creative design synthesis. *arXiv preprint arXiv:2103.06242*.

[Packard 1975] Packard, S. 1975. Aesthetics and psychobiology by de berlyne. *Leonardo* 8(3):258–259.

[Radford, Metz, and Chintala 2016] Radford, A.; Metz, L.; and Chintala, S. 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. *ICLR*.

[Reed et al. 2016] Reed, S. E.; Akata, Z.; Mohan, S.; Tenka, S.; Schiele, B.; and Lee, H. 2016. Learning what and where to draw. In *NIPS*.

[Ren et al. 2018] Ren, M.; Ravi, S.; Triantafillou, E.; Snell, J.; Swersky, K.; Tenenbaum, J. B.; Larochelle, H.; and Zemel, R. S. 2018. Meta-learning for semi-supervised few-shot classification. In *International Conference on Learning Representations*.

[Sbai et al. 2018] Sbai, O.; Elhoseiny, M.; Bordes, A.; LeCun, Y.; and Couprie, C. 2018. Design: Design inspiration from generative networks. In *ECCV workshop*.

[Tendulkar et al. 2019] Tendulkar, P.; Krishna, K.; Selvaraju, R. R.; and Parikh, D. 2019. Trick or treat: Thematic reinforcement for artistic typography. In *ICCC*.

[WikiArt 2015] WikiArt, O. 2015. Wikiart dataset. https://www.wikiart.org/. Accessed: 2020-05-30.

[Wundt 1874] Wundt, W. M. 1874. *Grundzüge der physiologischen Psychologie*, volume 1. W. Engelman.

[Zhang et al. 2017] Zhang, H.; Xu, T.; Li, H.; Zhang, S.; Wang, X.; Huang, X.; and Metaxas, D. 2017. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*.

[Zhang et al. 2018] Zhang, R.; Che, T.; Ghahramani, Z.; Bengio, Y.; and Song, Y. 2018. Metagan: An adversarial approach to few-shot learning. In *Advances in Neural Information Processing Systems*, 2371–2380.

Semantic AI models for guiding ideation in architectural design courses

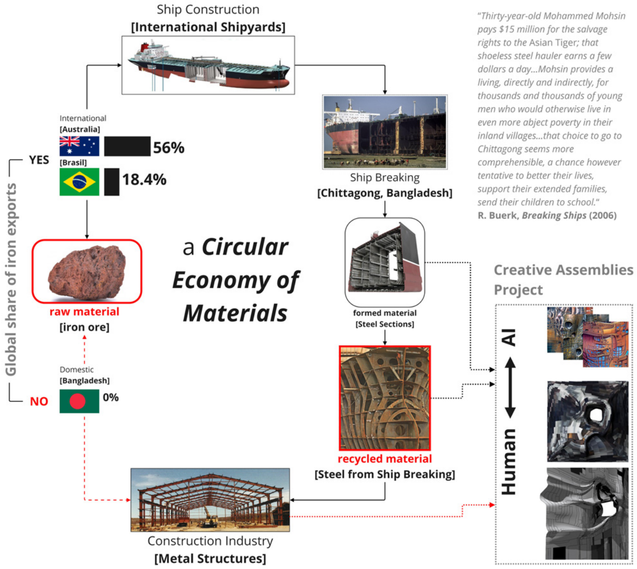
Emmanouil Vermisso
School of Architecture
Florida Atlantic University
111 East Las Olas Boulevard
Fort Lauderdale, FL 33301, USA
evermiss@fau.edu

Abstract

This paper discusses the combination of available artificial intelligence (AI) models, i.e. Neural Language Models (NLMs) with trained GANs and human interpretation to facilitate architectural ideation. The workflow identifies conceptual scenarios for a speculative design using semantic prompts. Results become visual references to complement revised semantic descriptions for guiding a VQGAN+CLIP model, leveraging control over the outcomes, which are then sorted using dimensionality reduction and further curated for training other models (GANs). The NLMs' interpretation of text input increases the possibility of spanning greater semantic distances, towards creative visual outcomes, while the nested workflow of AI-human steps enables an automated query of a larger solution space. Furthermore, it considers the problem of low-bandwidth, reductionist encoding of visual data (Hadamard, 1945) resulting from verbal-based (NLM) processing models (LeCun, 2021) that potentially constrain design agency.

1. Introduction

A reconsideration of the binary separation of intelligence as human and artificial, and the perception of intelligence as a "spectrum" (Bratton, 20 Feb.2022) may enhance human capacity to make decisions by introducing AI as an assistant for design creativity. This paper tries to propose a pedagogical structure where the agency of human designers is complemented by generative tools including language-based AI models (i.e.VQGAN+CLIP) for an architectural design studio course. The workflow involves a back-and-forth shift of "agency", trying to establish the type of framework where AI can make a contribution in relation to human-driven decision stages. Due to current limitations of Artificial Neural Networks (ANNs), and specifically Neural Language Models (NLMs), applying AI's potential in a heuristic manner seems more feasible than pursuing a clear problem within a deterministic design approach. Furthermore, NLMs' limitation when used in isolation warrants their integration with other ANNs (Bolojan, Vermisso, & Yousif, 2022).As a result, this project theme offers an open-ended, bottom-up way to craft a design agenda. The research involves the integration of computational generative tools with more analog design steps to re-imagine design scenarios for re-assembly of large, broken cargo ships and oil tankers into informal dwelling communities. The specific geographical and socio-cultural backdrop (Fig.1) is used to introduce NLMs and other ANNs during the early stages of ideation. In addition, we contemplate on further design tasks where AI may be useful, like the identification of ways to synthesize particular semantic features with each other. The project is inspired by the empirical creativity of ship-breaking communities in Bangladesh and operates on the fringe of the regular economy. It is viewed as an exercise that is guided by (human-computational) creativity, while being grounded by material and assembly constraints. Pedagogically, it offers a methodological exploration of design agency during a process of ideation and generation.



"Thirty-year-old Mohammed Mohsin pays $15 million for the salvage rights to the Asian Tiger; that shoeless steel hauler earns a few dollars a day...Mohsin provides a living, directly and indirectly, for thousands and thousands of young men who would otherwise live in even more abject poverty in their inland villages...that choice to go to Chittagong seems more comprehensible, a chance however tentative to better their lives, support their extended families, send their children to school."
R. Buerk, ***Breaking Ships*** (2006)

Figure 1 Iron supply for ship construction is recycled from unregulated ship-breaking (Bangladesh). The design project lives along the margins of this conceptual space, assisted by human and automated (AI) input.

Design Inquiry: The process of invention

The proposed workflow leverages generative models to operate in a heuristic fashion, allowing an expanded search in the solution space, for initial design conditions like visual inspiration. During this early stage, when a clear question or problem is absent, a flexible approach can trigger creative thinking. This argument has been widely discussed by experts including mathematician Jacques Hadamard, in his study of the cognitive mechanisms of mathematical invention, and the identification of a given problem: "*What* *shall we try to discover? What problem shall we try to solve?*" (Hadamard, 1945). Hadamard mentions neurologist Édouard Claparède's distinction between two kinds of invention: the former looks for the means to reach a particular goal (*question* to *solution*), while the latter imagines the usefulness of a fact that is discovered, possibly to be exploited much later (*answer before question*). Hadamard notes that human progress lies largely on the second kind. As Donald Rumsfeld famously remarked, besides what we know that we know and don't know, there exist "unknown unknowns", things which we would not even consider in our selection. (Zak, 2021) Adopting a flexible methodology opens the way to such unexpected discovery.

2. State-of-the-art: Architectural AI & NLM

Although Artificial Intelligence has begun to concern architects in the past 3-5 years for the most part, Makoto Sei Watanabe was one of the first architects to design using an AI interface in the late 1990s. His work used a number of inputs processed through an algorithm, which returned an output which was evaluated and scored by the designer, helping the algorithm to revise its performance. (Watanabe, 2005) Today, a number of architects use Generative Adversarial Networks (GANs) (Goodfellow, et al., 2014) because of their high quality results in computer vision, to perform data interpolation (StyleGAN) or domain transfer (CycleGAN) with impressive results (Bolojan, The Hitchhiker's Guide to Artificial Intelligence: AI and Architectural Design, 2021) Unfortunately, these kind of models can be computationally expensive, requiring precise curation of data and substantial computational resources, to reach very high resolutions. A type of network which has recently offered generative capacities through language are Neural Language Models, released in 2021. NLMs like CLIP, DALL-E and VQGAN+CLIP are pretrained on large datasets, so they are computationally cheap. (Rodrigues, Alzate-Martinez, Escobar, & Mistry, 2021) used VQGAN+CLIP for design inspiration, using text prompts in literal (analogy) and abstract (metaphorical) ways, combined with photographs and sketches. It is important to note that Watanabe mentioned, in his work, the separation of one architecture "condition" (i.e. form) from others, considering separate inputs within an AI-assisted workflow. This work aligns with this intuition, proposing to replace singular AI models with multiple ones which perform various tasks, including NLMs and GANs.

3. Design Methods

A design workflow is proposed, which includes manual curation of semantic descriptors, automated generation of visual spatial scenarios in 2D (Wombo "Dream" AI) and sorting of the visual outcomes (PixPlot), manual qualification of the results and a second layer of automated 2D scenarios generation (VQGAN+CLIP). The current state of this workflow (Fig.2) offers a robust method for generating an expanded search space of conceptual spatial references for further interrogation. Students used these methodologies as a foundation for developing a catalogue of design scenarios which were 3d modeled, implementing panelization strategies (GH+Ivy) to rationalize the surfaces and speculate how these could be constructed from reclaimed steel from ship-breaking. The discussion herewith will focus on the early stages of the catalogue material, and how this process can be refined using additional generative models beyond NLMs. Overall, this is a priori a speculative exercise to assess a process of varying agency, where the designer relinquishes control during certain parts of the process while he interferes elsewhere. Among the objectives is intuiting how to steer NLMs towards results which align with the design narrative, and identify how to use these automated steps in a meaningful way (i.e. producing nested "collage"-like drawings for concept generation.)

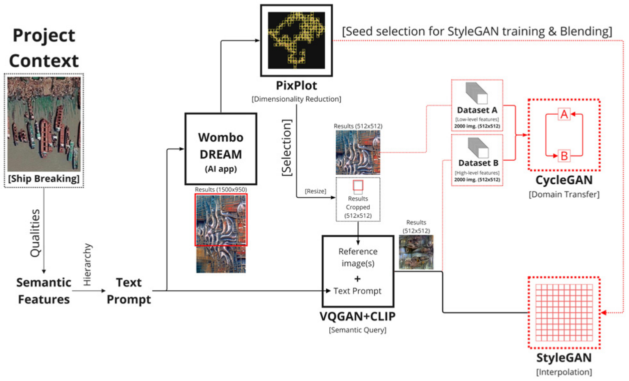


Figure 2. Workflow connecting NLMs with GANs.

NLMs for preliminary scenario generation

The "Dream" app by WomboAI is based on some internalized neural-language-model in the fashion of other existing language-based AI models like "DALL-e" or "VQGAN+CLIP", which contain a deep artificial neural network (i.e. GAN) for classification of visual data based on linguistic correlation and one for generation of visual results based on the classification. Such AI models can generate fairly complex visual results from text prompts of varying specificity. It is unquestionable that this type of results are seductive due to their visual complexity and speed of generation. However, it is important to question their significance for design decision making. As far as usefulness goes, the designer's agency is minimized, as it is difficult to intervene within the language-based AI model (in this case, Wombo's "Dream" app). Although we cannot accept these AI-generated outcomes as direct valid steps in architectural designing, they are likely references in the act of ideation, helping inspire and steer our design inquiry towards certain directions which may have otherwise remained latent. To efficiently manage the large number of results which can quickly accumulate from these automated processes, it is important to categorize the properties (spatial, tectonic, visual, formal etc.) which qualify a scenario as interesting, successful and/or optimal.

The creation of separate categories which relate to the various semantic descriptors is necessary to identify areas of interest in the visual results. Naturally, every outcome displays interesting high and low-level features. (Fig.3)

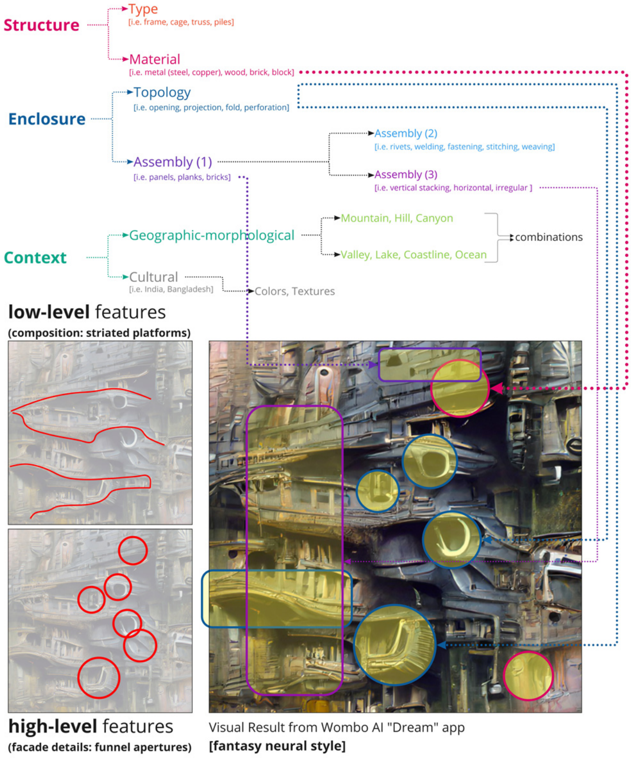

Structure
Type
[i.e. frame, cage, truss, piles]
Material
[i.e. metal (steel, copper), wood, brick, block]
Enclosure
Topology
[i.e. opening, projection, fold, perforation]
Assembly (1)
[i.e. panels, planks, bricks]
Assembly (2)
[i.e. rivets, welding, fastening, stitching, weaving]
Assembly (3)
[i.e. vertical stacking, horizontal, irregular]
Context
Geographic-morphological
Mountain, Hill, Canyon
Valley, Lake, Coastline, Ocean
combinations
Cultural
[i.e. India, Bangladesh]
Colors, Textures
low-level features
(composition: striated platforms)
high-level features
(facade details: funnel apertures)
Visual Result from Wombo AI "Dream" app
[fantasy neural style]

Figure 3. Schematic structure to identify correlation between design aspects and semantic feature description (text prompt); High and low-level image features.

It is not important to establish an ideal scenario, but to extrapolate qualities which can be translated into inspiring and applicable design features, i.e.strategies to create openings on a large surface by *perforating, folding, bulging, projecting*. Combining particular features should also be considered. Figure 3 gives an example of "high" and "low" level features. As a note, the same characteristic (i.e. a certain type of opening shape) can be read as a high-level feature in an image or a low-level feature in another, depending on its scale relative to the overall composition. In this paper, we will refer to properties like overall compositional structure as a "low-level" feature while "high-level" features will identify finer details inside a composition (openings, textures, individual structural members, etc). While humans are good at recognizing patterns, sorting through large samples of data with thousands of features requires another interface. In order to sort the AI-generated images form the NLMs (stage 1), we used dimensionality reduction (UMAP) and clustering algorithms (K-means) in PixPlot (Fig.4). The data comprised 861 images, grouped into 10 clusters based on their characteristics. It was clear that the network organized results from the same text prompt or same neural style (option in Wombo AI) together, picking high-level features like *woven configurations, bulging topologies, pleating,* as parameters for clustering, as well as other more visual ones like color distribution. Based on the PixPlot matrix, results with certain complex features were selected as reference images to guide a VQGAN+CLIP model. We are currently looking at selecting particular successful seeds from radically different clusters for blending through StyleGAN later, because blending seeds from different feature families may output interesting results.

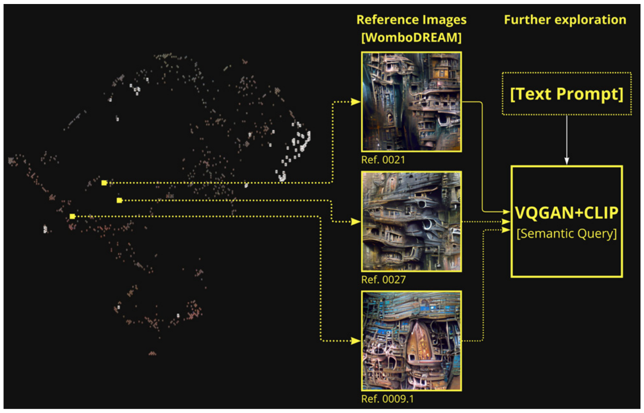


Figure 4. Visualization of 861 results (left) from the "Dream" app, sorted in 10 clusters, using PixPlot.

4. Results & Discussion: Encoding intentions

According to experts, words -like other aspects of our existence- are gradually "domesticated", entering a fully conscious state of acceptance via "methodical selection" and selection for regular use depending on our preference. (Dennett, 2017) A characteristic example of word types which are not "preferred", but necessary, are technical terms in a scientific field, which are commonly accepted to describe the field itself. In architecture, an array of semantic descriptors has been consciously adopted when referring to design properties or construction attributes. Selecting such semantic references to guide NLMs, i.e. Wombo "Dream" or VQGAN+CLIP is normal, leading to visual outcomes which are "expected", typically reflecting "realistic" qualities which the intended spatial configurations need to possess, to qualify as successful. If we are looking for something unique, creative, the choice of words needs to extend beyond the conscious selection of descriptors which are semantically close to each other and typical to the primary reference context (ship breaking), to ones which are not typically combined. We have tried to work with text prompts which supplement the 'expected' features (i.e. *metal ship hull; slum dwelling,* etc.) with ones which are unfamiliar and semantically distant (i.e. *fighter jet intake; woven exoskeleton*) (Fig.5) Increasing the semantic distance is an obvious idea; regarding the notion of "relevance", Boden mentioned that it is difficult to traverse large conceptual distances, but it is also more likely to reach novel concepts. (Boden, 2013) We tried to "lend" our human intuition to the network's search descriptors via unusual prompts, to inquire what results might be obtained.

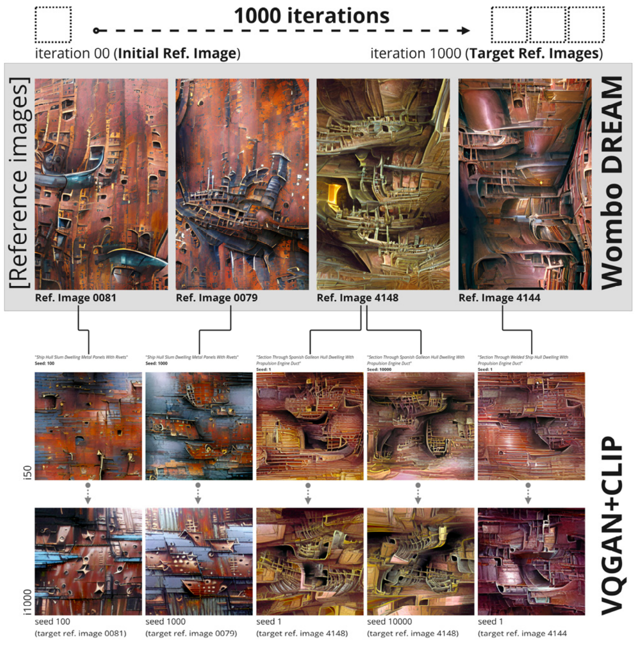

1000 iterations
iteration 00 (Initial Ref. Image)
iteration 1000 (Target Ref. Images)
[Reference images]
Wombo DREAM
Ref. Image 0081
Ref. Image 0079
Ref. Image 4148
Ref. Image 4144
"Ship Hull Slum Dwelling Metal Panels With Rivets"
Seed: 100
"Ship Hull Slum Dwelling Metal Panels With Rivets"
Seed: 1000
"Section Through Spanish Galleon Hull Dwelling With Propulsion Engine Duct"
Seed: 1
"Section Through Spanish Galleon Hull Dwelling With Propulsion Engine Duct"
Seed: 10000
"Section Through Welded Ship Hull Dwelling With Propulsion Engine Duct"
Seed: 1
i50
i1000
seed 100
(target ref. image 0081)
seed 1000
(target ref. image 0079)
seed 1
(target ref. image 4148)
seed 10000
(target ref. image 4148)
seed 1
(target ref. image 4144
VQGAN+CLIP

Figure 5. Wombo Dream results guide VQGAN+CLIP.

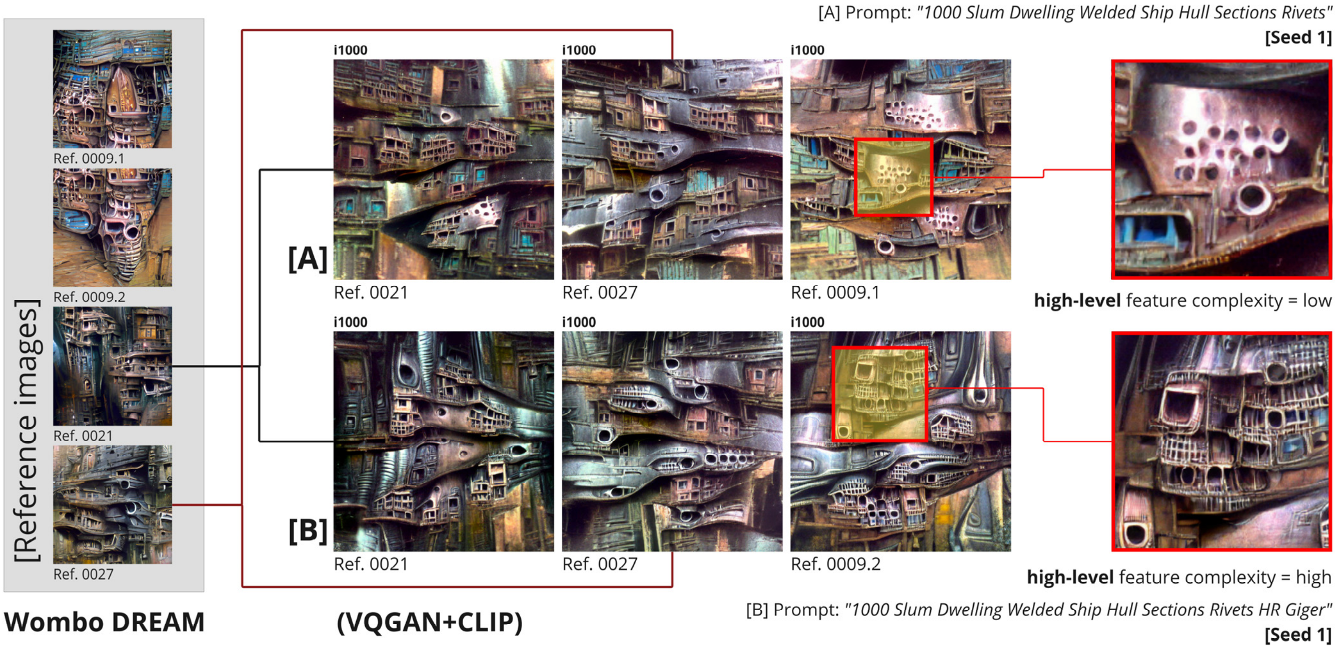

[A] Prompt: "1000 Slum Dwelling Welded Ship Hull Sections Rivets"
[Seed 1]
[Reference images]
Ref. 0009.1
Ref. 0009.2
Ref. 0021
Ref. 0027
i1000
i1000
i1000
[A]
Ref. 0021
Ref. 0027
Ref. 0009.1
high-level feature complexity = low
i1000
i1000
i1000
[B]
Ref. 0021
Ref. 0027
Ref. 0009.2
high-level feature complexity = high
[B] Prompt: "1000 Slum Dwelling Welded Ship Hull Sections Rivets HR Giger"
[Seed 1]
Wombo DREAM
(VQGAN+CLIP)

Figure 6. Scenarios generated with VQGAN+CLIP curate qualities consistent with the project theme: ship hull population with novel topological-tectonic features.

Fig.6 shows 6 results from a text prompt in VQGAN+CLIP with a target reference image; (one of four images -shown- which had been generated with Wombo was selected). The prompt for the first 3 trials was: "*1000 Slum Dwelling Welded Ship Hull Sections Rivets*" and the chosen seed was #1. As the reference images were generated in Wombo with a prompt including the phrase "HR Giger", we see the influence of that graphic language at a high-level feature is evident (but not exaggerated) in the results from the VQGAN+CLIP (A0021, A0027, A0009.2). We tried 3 more tests with the same prompt, adding "HR Giger" at the end of the prompt and keeping the seed to #1 ("*1000 Slum Dwelling Welded Ship Hull Sections Rivets HR Giger*"). It is clear that using the same semantic reference as the graphic reference in the target image is perhaps an exaggeration, because the high-level semantic features which relate to this formal language (the low-level arrangement does not seem to vary much) will be searched for both via language reference and image reference, resulting in overpowering visual results. (B0021, B0027, B0009.1) The image reference of this feature is sufficient to assign the preferable, as it works in a stronger sense than the text prompt, based on the results herewith shown (A/B009, A/B0021, A/B0027). However it should be noted that the results from the (B) trials demonstrate -despite their exaggerated "Giger" style, interesting details at the high-level resolution. Whether a semantic feature is pursued via both target image and prompt reference depends, therefore, on the specific scenario being considered.

Future Work: Feature Disentanglement in GANs

Due to the intrinsic limitations of language, introducing additional AI models is warranted to explore detailed semantic combinations further. We propose AI models which are good at interpolation, to combine known elements into new configurations (Boden, 2013). We are in the process of training a StyleGAN network, to later blend a number of qualified new seeds with selected features, as well as existing seeds from Wombo Dream. StyleGAN can perform feature disentanglement so these can be reconfigured into new organizations with emergent visual-spatial qualities.

Author Contributions

[EV] ideated and wrote the paper alone.

Acknowledgments

The author would like to thank his Architectural Design 10 students for their willingness to work with tools they were entirely unfamiliar with. The author would also like to thank his colleagues at FAU, Daniel Bolojan and Shermeen Yousif, for the broader research discussions which eventually led to this line of work. The author also acknowledges the research team at Yale Digital Humanities Lab, where PixPlot was developed, the people behind the WomboAI “Dream” app, and Katherine Crowson for making the VQGAN+CLIP model available.

References

Boden, M. (2013). Creativity as a Neuroscientific Mystery. In O. Vartanian, A. S. Bristol, & J. C. Kaufman, Neuroscience of Creativity (pp. 3-18). Cambridge, MA: MIT Press.

Bolojan, D. (2021, June 20). The Hitchhiker’s Guide to Artificial Intelligence: AI and Architectural Design. Retrieved from www.digitalfutures.world: https://www.youtube.com/digitalfuturesworld/live

Bolojan, D., Vermisso, E., & Yousif, S. (2022). Is Language All We Need? A Query Into Architectural Semantics Using a Multimodal Generative Workflow. CAADRIA 2022: Post-Carbon. Sydney: CAADRIA.

Dennett, D. C. (2017). From Bacteria to Bach and Back: The Evolution of Minds. New York: Penguin Books.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., . . . Bengio, Y. (2014, June 10). Generative Adversarial Nets. Retrieved from https://arxiv.org: https://arxiv.org/pdf/1406.2661.pdf

Hadamard, J. (1945). The Mathematician’s Mind: The Psychology of Invention in the Mathematical Field. Princeton: Princeton University Press.

Rodrigues, R. C., Alzate-Martinez, F. A., Escobar, D., & Mistry, M. (2021). Rendering Conceptual Design Ideas with Artificial Intelligence: A Combinatory Framework of Text, Images and Sketches. ACADIA 2021: Realignments: Towards Critical Computation.

Watanabe, M. S. (2005). Algorithmic Design/Induction Design: Three Kinds of Flow/Three Stations. Retrieved from https://www.makoto-architect.com: https://www.makoto-architect.com/kashiwanohaCSt.html

Zak, D. (2021). ‘Nothing ever ends’: Sorting through Rumsfeld’s knowns and unknowns. Retrieved February 23, 2022, from https://www.washingtonpost.com/lifestyle/style/rumsfeld-dead-words-known-unknowns/2021/07/01/831175c2-d9df-11eb-bb9e-70fda8c37057_story.html

Seeding Diversity into AI Art

Marvin Zammit, Antonios Liapis and Georgios N. Yannakakis
Institute of Digital Games, University of Malta, Msida MSD2080, Malta
{marvin.zammit,antonios.liapis,georgios.yannakakis}@um.edu.mt

Abstract

This paper argues that generative art driven by conformance to a visual and/or semantic corpus lacks the necessary criteria to be considered creative. Among several issues identified in the literature, we focus on the fact that generative adversarial networks (GANs) that create a single image, in a vacuum, lack a concept of novelty regarding how their product differs from previously created ones. We envision that an algorithm that combines the novelty preservation mechanisms in evolutionary algorithms with the power of GANs can deliberately guide its creative process towards output that is both good and novel. In this paper, we use recent advances in image generation based on semantic prompts using OpenAI's CLIP model, interrupting the GAN's iterative process with short cycles of evolutionary divergent search. The results of evolution are then used to continue the GAN's iterative process; we hypothesise that this intervention will lead to more novel outputs. Testing our hypothesis using novelty search with local competition, a quality-diversity evolutionary algorithm that can increase visual diversity while maintaining quality in the form of adherence to the semantic prompt, we explore how different notions of visual diversity can affect both the process and the product of the algorithm. Results show that even a simplistic measure of visual diversity can help counter a drift towards similar images caused by the GAN. This first experiment opens a new direction for introducing higher intentionality and a more nuanced drive for GANs.

Introduction

Visual art is among the most well-researched domains in computational creativity as it is perhaps the most recognisable among tasks which, when performed by humans, are deemed creative (Ritchie 2007). Painting in any style or medium requires some degree of skill (Colton 2008), and endowing machines with painting skill has a long and exciting history (Cohen 2017; Colton 2012; Lindemeier et al. 2015; Machado and Cardoso 2002). A watershed moment in this endeavour has been the advent of Generative Adversarial Networks (GANs) (Goodfellow et al. 2014), which not only started to bridge the gap between human and machine performance but also allowed novices to generate compelling images without extensive technical knowledge, development effort, or access to specialised hardware. Generative art produced through deep learned models has taken the world by storm in the last five years. The strength of models trained in vast image databases in producing highly typical content, such as human faces, has led to an almost ubiquitous fascination by researchers, artists, laymen, media, and speculators. We follow McCormack, Gifford, and Hutchings (2019) and refer to visuals generated via deep learning as "AI Art" in this paper.

As the general public became more interested in AI Art, a crucial component for the perception of creativity hinged on whether the software could explain in natural language the framing information regarding what it was trying to portray (Colton, Charnley, and Pease 2011). While several GAN architectures addressed the generation of images from text prompts (Reed et al. 2016; Zhang et al. 2017), they performed well only in limited datasets and could not scale to generate visuals based on broader themes. The recent introduction of OpenAI's Dall-E (Ramesh et al. 2021) demonstrated an unprecedented high correspondence between a given text prompt and the generated image on different prompts. While neither the Dall-E model nor the training dataset have been publicly released at the time of writing, a pre-trained model of Contrastive Language-Image Pretraining (CLIP) is available (Radford et al. 2021). The release of CLIP energised researchers and enthusiasts alike, leading to many open-source projects and twitter bots that take advantage of the links between semantics and images to produce more convincing AI Art, such as album titles and covers[1].

In the context of computational creativity, however, it would be easy to argue that images generated only to conform to the patterns of the corpus fall into "mere generation" (Ventura 2016) and lack authenticity (McCormack, Gifford, and Hutchings 2019). Using the criteria of *novelty*, *quality* and *typicality* regarding products of a creative process (Ritchie 2007), we argue that GANs and similar architectures target only typicality by conforming to patterns discovered in their training corpus. While we appreciate that there are several issues—such as intent and attribution (McCormack, Gifford, and Hutchings 2019)—that AI Art should address before it can be considered creative, we focus in this paper on the novelty of the product by endowing the algo-

[1] `https://twitter.com/ai_metal_bot`

rithm with a way to assess and prioritise diversity in its generated output.

While a product's novelty can be assessed in terms of past artefacts of the same type, we focus instead on contemporaneous novelty in a population of artefacts that are generated—iteratively—at the same time. While GANs are typically applied to generate a single image, our study also tests how diverse a population of images produced by GANs can be when the initial seeds are different. We take advantage of evolutionary algorithms that perform quality-diversity search (Pugh, Soros, and Stanley 2016) and combine them with the power of deep learning through *cycles of exploration and refinement*. Taking advantage of trained models of semantic-image similarities, we test this process of iterative refinement (Liapis, Yannakakis, and Togelius 2013) when generating sets of images for five text prompts. This first experiment raises a number of questions regarding e.g. how image novelty can be assessed, and we test two different image metrics as both evolutionary goals and for analysing the quality of the final results.

Background Technologies

The proposed methodology combines complex, cutting-edge technologies of deep learning and divergent evolution. The relevant technologies and a high-level overview of their inner workings are presented below.

OpenAI CLIP

OpenAI's CLIP is a supervised neural network architecture which associates images with corresponding text and vice versa, learning underlying concepts within each of these domains (Radford et al. 2021). CLIP was released in January 2021 and quickly became popular for a wide variety of tasks, such as image classification (Esmaeilpour et al. 2021), semantic image generation (Ramesh et al. 2021), and captioning (Mokady, Hertz, and Bermano 2021).

CLIP is essentially a zero-shot classifier, which was pre-trained using images and corresponding textual phrases scraped from the internet. The training dataset itself was not released but it contained $4 \cdot 10^8$ text-image pairs. The structure of CLIP consists of a Transformer-based model (Vaswani et al. 2017) which encodes the tokenised input text batches. For the image encoding, two different architectures were compared; a ResNET-D based model (He et al. 2019) and a Vision Transformer (ViT) model (Dosovitskiy et al. 2021). Batches of image-text pairs are encoded and cross-processed using contrastive learning (Van den Oord, Li, and Vinyals 2018) in order to train the model to predict the probability that a given text input matches a given image or vice versa. The resulting trained models matched or outperformed some of the best classifiers when applied to broad datasets, but ranked worse on specific, narrow-domain datasets. The benefit of CLIP in the current work is that it can provide a singular cosine similarity score (we refer to this as CLIP score in this paper) between a textual prompt and an image, for any semantic prompt. This CLIP score has been used to assess generated images and predetermined text input, and thus to steer various methods of GAN image generation towards some predetermined text input (Gal et al. 2021; Kim and Ye 2021). These CLIP-guided image generation experiments are often performed by enthusiasts and are not published; however, many early contributions are available in online repositories[2].

In practice, CLIP-guided image generation starts from a random fractal noise array as an image, and uses CLIP to generate its embedding. CLIP is also used to embed the input text prompt and the two sets of vectors are compared using cosine similarity, given by:

$$similarity(\vec{t}_{image}, \vec{t}_{prompt}) = \frac{\vec{t}_{image} \cdot \vec{t}_{prompt}}{|\vec{t}_{image}| \cdot |\vec{t}_{prompt}|} \quad (1)$$

where $\vec{t}_{image}$ and $\vec{t}_{prompt}$ are the CLIP vector embeddings of the image and the text prompt respectively, and $|\vec{t}|$ denotes the magnitude of vector $\vec{t}$.

Generative Adversarial Networks

Generative Adversarial Networks (GANs) were introduced by Goodfellow et al. (2014) and have since become an important milestone in AI Art. GANs consist of a generator which learns to generate artefacts within its domain, and a discriminator network which learns to distinguish between what the generator creates versus real artefacts. The output of the discriminator is used to train the generator, pitting their progress against each other, and resulting in greatly enhanced performance compared to previous techniques. Generally, the discriminator is discarded after training and only the generator is used for inference. This technique has been used extensively across different domains, for text to image generation (Brock, Donahue, and Simonyan 2018), style transfer (Karras, Laine, and Aila 2019), super-resolution upscaling (Ledig et al. 2017), and many more applications.

Vector Quantized Variational Autoencoders (VQVAE) are autoencoders which operate on image segments instead of individual pixels (Esser, Rombach, and Ommer 2021). Their networks combine convolutional layers with transformer structures, capturing short-range feature interactions with the former and long-range ones with the latter. An image at the encoder input is converted into a sequence of segments which are stored in a discrete code book of representations. An image is thus compressed to a sequence of indices representing the position of each segment within the code book.

During VQVAE training, a GAN architecture is used (often referred to as a VQGAN) to learn the weights and biases of the encoding and decoding networks, and also to determine the code book entries which will be available for these processes. Therefore, the training data has a significant impact on the variety of images which can be encoded or decoded by a VQGAN. Specifically, images with features that bear the closest resemblance to the training data set will be compressed and decompressed more faithfully

[2] `https://github.com/lucidrains/big-sleep` and `https://colab.research.google.com/drive/1L8oL-vLJXVcRzCFbPwOoMkPKJ8-aYdPN`, among others.

than images in a different domain. As discussed in the introduction, products of VQGANs therefore target *typicality* (Ritchie 2007) with the training set above all else.

VQGANs enable an easy translation between an image and its latent vector representation, offering a way to manipulate images that can be combined with both CLIP evaluation and latent vector evolution. By applying backpropagation to the latent vector conditioned by the CLIP score of its corresponding image to a given text prompt, an image can be directed towards a suitable representation for the latter.

Novelty Search with Local Competition

Evolutionary computation has a long history and has proven to be powerful in numerical optimisation tasks (De Jong, Fogel, and Schwefel 1997). However, it often struggles to discover appropriate individuals that can act as stepping stones for further improvements towards an objective. In deceptive fitness landscapes, such individuals may perform poorly in terms of the objective but may possess the necessary genotypic structure that can lead to highly fit individuals after a number of genetic operators are applied. Novelty as the (sole) evolutionary objective was introduced "as a proxy for stepping stones" (Lehman and Stanley 2008). Novelty search has shown great promise in many application domains such as robotics (Lehman and Stanley 2008; 2011), game content generation (Liapis, Yannakakis, and Togelius 2015; 2013; Liapis et al. 2013) and generative art (Lehman and Stanley 2012). While most publications in this vein apply novelty search to neuroevolution (Stanley and Miikkulainen 2002), it can also be applied to other types of indirect (Liapis 2016; Liapis et al. 2013) or direct embryogenies (Liapis, Yannakakis, and Togelius 2015).

Emulating evolutionary processes in nature, applying local competition to the process of natural selection showed greater promise in some applications (Lehman and Stanley 2011). Local competition pits individuals against phenotypically similar individuals in the search space. This Novelty Search with Local Competition (NSLC) allowed diverse features to survive and serve as stepping stones towards better specimen, even if their performance was only optimal locally. It empowered individuals with diverse features to survive and evolve without being overpowered by better developed features in other individuals. In practice, NSLC operates as a multi-objective optimisation problem where one objective is increasing the *novelty score* and the other objective is increasing the individual's *local competition score*. Both scores compare an individual with its nearest neighbours in a behavioural space; these neighbours may be from the current population or from an archive of novel past individuals. The novelty archive is populated during evolution, with the most novel individuals in every generation being added to the archive. The novelty score is calculated via Eq. (2), as the average distance of this individual with its nearest k neighbours. The local competition score is calculated via Eq. (3), as the ratio of nearest k neighbours that the individual outperforms in terms of fitness. Evidently, the algorithm hinges on two important parameters: the *fitness metric* which affects the local competition score, and the *distance metric* which affects the nearest neighbours being considered and the novelty score in general.

$$n(i) = \frac{1}{k}\sum_{j=1}^{k} d(i, \mu_j) \tag{2}$$

$$lc(i) = \frac{1}{k}\sum_{j=1}^{k} o_f(i, \mu_j) \tag{3}$$

where $d(x, y)$ is the behavioural distance between individuals x and y and depends on the domain under consideration, μ_j is the j-th nearest neighbour to i, $o_f(x, y)$ is 1 if $f(x) > f(y)$ and 0 otherwise, where f is the fitness function for the current problem. The current population and the novelty archive are used to find the nearest neighbours.

Proposed Methodology

At its core, our proposed methodology revolves around an alternating sequence of refining cycles (via backpropagation) and exploration cycles (via divergent evolution). Using CLIP-guided VQGANs in this instance, we describe the specific methods followed in each cycle below.

Backpropagation Cycle

In order to take advantage of the power of GAN architectures in converting seemingly random noise into visually appealing images, we use backpropagation-driven cycles to start the process and as a final step of refining an image before showing it to a human audience.

The code used for semantic image generation is based on Pixray[3], using pixel-based generation through VQGANs (Esser, Rombach, and Ommer 2021). Details of the VQGAN technologies are in the Background section. For this paper we adopt a VQGAN pre-trained on the WikiArt dataset (Tan et al. 2016). WikiArt is a dataset of $81,444$ images of artistic creations (paintings, images) across many different art styles[4]. The images produced from the WikiArt-trained VQGAN are more illustrative and surreal rather than representational or photorealistic, which suits our goals of producing artefacts that observers would consider creative. Moreover, the images generated for each prompt were deemed to be more visually similar to each other than other models, when starting from different random seeds.

The generated images have dimensions of 384 by 384 pixels, and the VQVAE model sectioned the images into blocks of 16 by 16 pixels, resulting in a latent vector of 576 integers, each representing an index of the code book entry used to represent that block. Each integer's value range is $[0, 16384]$, as part of the autoencoder's code book.

At the start of the experiment, we randomise each latent vector using a random fractal noise array, and use CLIP to generate its embedding. In subsequent iterations, we use the negated CLIP's cosine similarity of Eq. (1) as a loss function to guide the backpropagation process towards a latent vector which produces an image that better matches the semantic

[3] `https://github.com/pixray/pixray`

[4] `https://archive.org/details/wikiart-dataset`

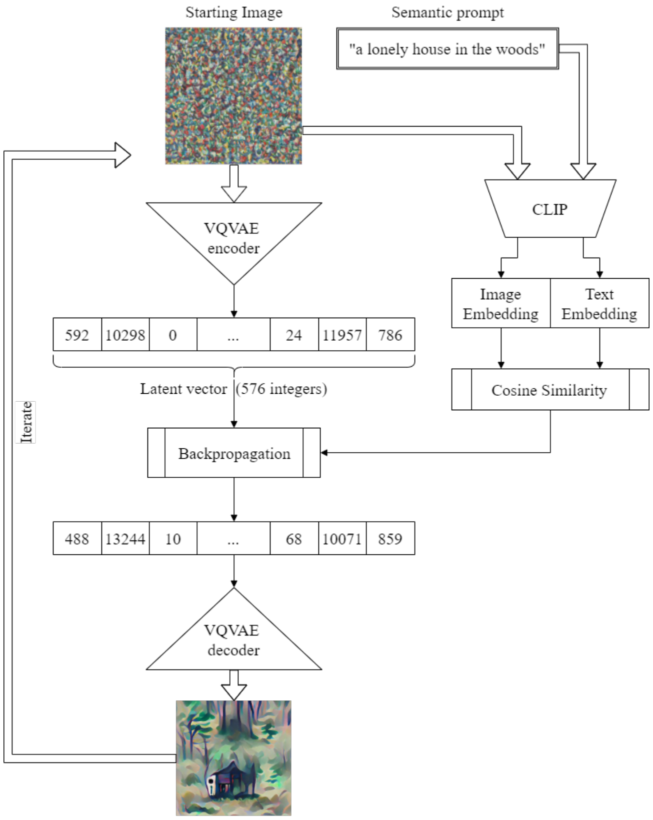


Figure 1: GAN iterations guided by CLIP.

prompt. Since the latent vector consists of integers and is not compatible with the continuous requirement for gradient descent, the internal tensor representation of the vector within VQGAN (consisting of floating point numbers) is used to backpropagate the CLIP loss. Fig. 1 visualises this process.

Exploration Cycle

Exploration cycles are carried out via novelty search with local competition (NSLC) operating on the latent vector representing the image. The genotype (latent vector) consists of 576 integers, ranging between 0 and 16384. Since each gene is an integer that is mapped in a very indirect way to some image segment, the evolutionary algorithm uses only mutation operators. In each mutation, 5% of the individual's genes (chosen randomly) are replaced with random integers between $[0, 16384]$. This mutation rate was chosen based on initial trials, as it can create perceptible perturbations in the image without making it unrecognisable within one application of mutation (as is the case with higher mutation rates).

For this experiment, $k = 15$ nearest individuals are considered for calculating both the novelty score and the local competition (LC) score, as per Eq. (2)-(3). In each generation, the $e = 3$ most novel individuals are added to the novelty archive. Note that the novelty archive starts empty at the start of each exploration cycle; there is no carryover from previous exploration cycles. The archive growth as the algorithm progresses increases the computational requirements, so this strategy of always adding a few individuals offered a good compromise between benefit and performance.

A Non-dominated Sorting Genetic Algorithm (NSGA-II) (Deb et al. 2002) was used to process the resulting two metrics (novelty and LC score) as a multi-objective optimisation problem, using the Pymoo Python library (Blank and Deb 2020). A minimal Pareto front is calculated for the two objectives; individuals closest to this front are dominant over the remaining population and are selected for the next generation. If more individuals are required after exhausting those on the Pareto front, a next best Pareto front is calculated and a next set of individuals is selected from therein. If there are more individuals on the Pareto front than those required to survive, then individuals are selected to create sparsity in the objective space. The sparsity is based on the Manhattan distance between individuals within this space.

One of the major challenges in this work was defining diversity in the generated images for the purposes of NSLC. As noted in the Background section, the behavioural distance affects which neighbours are considered for both novelty and LC, and in turn affects how we envision novelty in the final product (Ritchie 2007). It is a relatively easy task for a human to identify visual similarity between two images, but there are several challenges in quantifying similarity or diversity into a simple metric. In this work we compare image novelty by using two different approaches:

Chromatic Diversity (HSV): With this approach, we hypothesise that the distribution of colours in the pixels reflect the diversity of the images (Machado et al. 2015). We consider the hue, saturation and brightness of each pixel and for any two generated images I_1 and I_2 we derive a diversity metric from their means and standard deviations as follows:

$$m_1 = \Delta\overline{b} = |\overline{b}_1 - \overline{b}_2| \tag{4}$$

$$m_2 = \Delta\sigma(b) = |\sigma(b_1) - \sigma(b_2)| \tag{5}$$

$$m_3 = \Delta\overline{s} = |\overline{s}_1 - \overline{s}_2| \tag{6}$$

$$m_4 = \Delta\sigma(s) = |\sigma(s_1) - \sigma(s_2)| \tag{7}$$

$$m_5 = \Delta\overline{h} = |min[\overline{h}_1 - \overline{h}_2, \overline{h}_1 - (1 - \overline{h}_2)]| \tag{8}$$

$$m_6 = \Delta\sigma(h) = |\sigma(h_1) - \sigma(h_2)| \tag{9}$$

where h, s and b denote the hue, saturation and brightness, the means $(\overline{h}, \overline{s}, \overline{b})$ are taken across all the pixels in I_1 and I_2, and σ denotes the standard deviation of these values. Note that since the hue value is cyclic, its mean and standard deviation were calculated as follows:

$$\overline{h} = tan^{-1}\left(\frac{\sum_{i=1}^{N} sin(h)}{\sum_{i=1}^{N} cos(h)}\right) \tag{10}$$

$$\sigma(h) = \sqrt{\frac{\sum_{i=1}^{N}(min(h_i - \overline{h}, h_i - (1 - \overline{h})))^2}{N - 1}} \tag{11}$$

where N is the number of pixels in the image.

All h, s, b values are normalised in the $[0, 1]$ value range before the above calculations. We calculate the distance metric d_{HSV} as the mean square value of the individual metrics $m_1 \ldots m_6$.

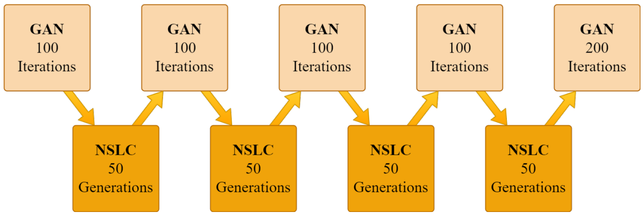


Figure 2: Structure of the experiments alternating between GAN and evolutionary NSLC cycles.

Visual Transformer Diversity (ViT): Another way to assess diversity is based on the embeddings of pre-trained models. Transformers (Vaswani et al. 2017) have shown an outstanding performance when applied to image classification (Dosovitskiy et al. 2021; Wu et al. 2020). Within its layers, the model encodes information about different images in its training set, and uses it to discern different images. We utilise this encoded information with a ViT model pre-trained on the ImageNet data set (Deng et al. 2009), and stripping its last layer. Since the last layer of ViT is used for image classification, by removing it we retain a latent vector of 768 floating point values for each processed image. We calculate a value of diversity (d_{ViT}) by taking the Euclidean distance between the latent vectors of two images.

Experiment

In order to assess how our envisioned algorithm that combines latent variable evolution (Bontrager et al. 2018) towards novelty with GANs, the following section reports our findings when producing novel sets of images for different semantic prompts. We first describe our choice of prompts and parameter setup, followed by a quantitative analysis of both the process and the final product, and conclude with a qualitative view of the resulting images.

Protocol

For the purposes of demonstrating our proposed methodology in a visual creativity task, we use the Pixray image generation system which leverages pretrained VQVAE models. Importantly, we wish to explore how the method operates in a variety of settings while still being able to compare with existing research. To facilitate this, we test five semantic prompts (SP) used by the community[5]:

- a lonely house in the woods (SP1)
- a pyramid made of ice (SP2)
- artificial intelligence (SP3)
- cosmic love and attention (SP4)
- fire in the sky (SP5)

For this experiment, we generate a population of 50 images by running Pixray for a total of 600 iterations. The initial population consists of latent vectors encoded from a set of randomly generated fractal noise images. The same initial population of images is used in all tested variations of our algorithms, across all prompts. To establish our GAN baseline (GAN-BSL), we run the process uninterrupted for each initial latent vector for 600 iterations in order to collect the final population. Initial experiments showed that at 600 iterations the composition of the image is stable, and although more iterations will refine it, the image does not change much. For our NSLC experiments, we interrupt the GAN process after 100, 200, 300 and 400 iterations and take the latent vectors of the images at that point to produce an initial population for NSLC; NSLC evolves for 50 generations, guided by either ViT (NSLC-ViT experiment) or HSV (NSLC-HSV experiment) distance metrics, and the final evolved population is then used to continue the GAN process (until interrupted again). The process is clarified in Figure 2.

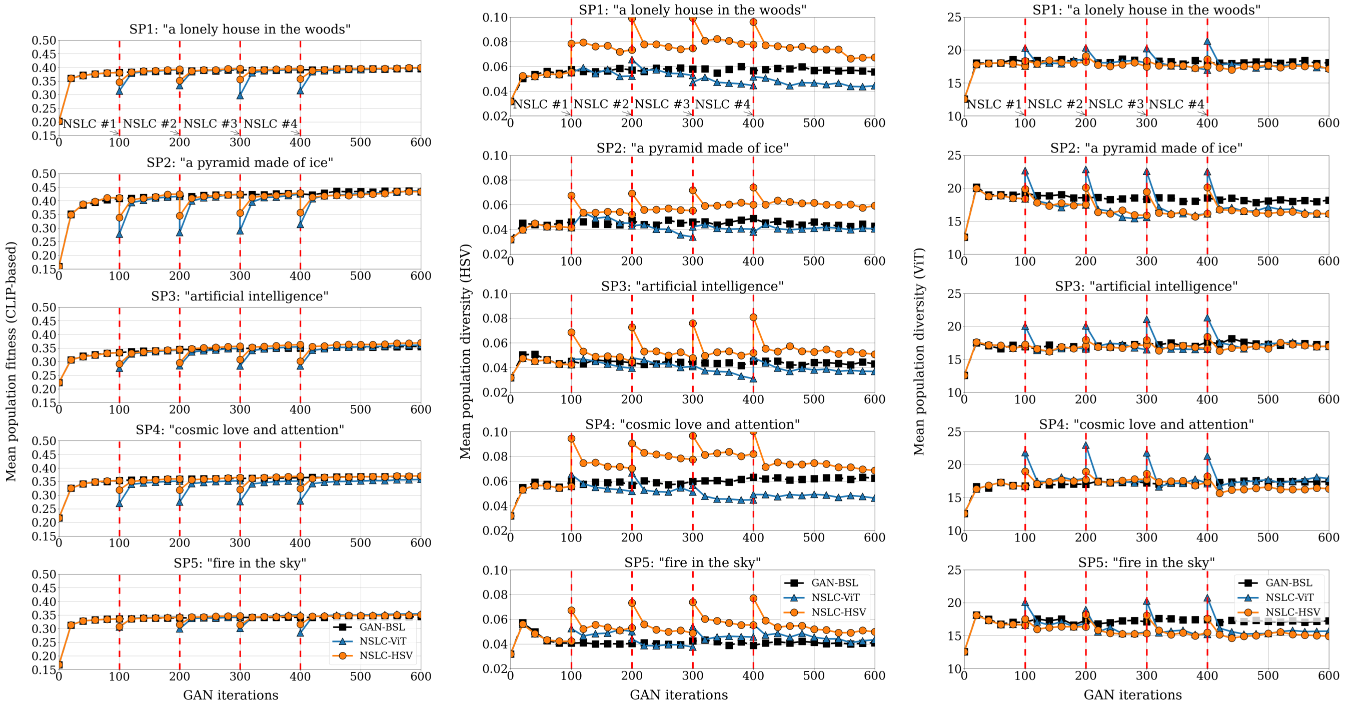
Evaluating the novelty or quality of the generated output is not straightforward (Ritchie 2007). For the purposes of this paper, we align these notions with the quality-diversity characterisations of NSLC and use the following performance metrics to compare the different algorithms:

- **mean fitness** based on the CLIP score across all 50 images in the population.
- **mean ViT diversity** calculated as the average ViT distance from the nearest 15 neighbours per individual, averaged across all 50 images in the population. Note that for this metric only the current population is considered for finding nearest neighbours (no archive).
- **mean HSV diversity** is calculated identically to mean ViT novelty using the HSV metric for measuring distance and finding nearest neighbours.

Numerical Results

We are equally interested in the *process* followed by the algorithms tested as we are in the *product* at the end of 600 iterations (Jordanous 2016). Therefore, Figure 3a shows how the mean fitness (CLIP score) fluctuates at different GAN iterations. Evidently, with the uninterrupted GAN-BSL the algorithm increases its accuracy quickly in the first 20 iterations but then continues to slowly improve. When the process is interrupted by NSLC cycles, the evolved population's fitness drops by 12% on average for NSLC-HSV and by 21% for NSLC-ViT. Surprisingly, the drop is nearly as substantial when NSLC is applied at later iterations, even if the (seed) images are well-formed at that point. It is evident that after each NSLC cycle, the GAN has a similar behaviour as when facing random initial seeds and can quickly restore the CLIP score to a similar level as the GAN-BSL at the same iteration (before quickly dropping again at the next NSLC cycle). At the end of the 600 iterations, all three algorithms seem to be reaching a very similar mean fitness score, although in almost all cases both NSLC variants reach slightly higher scores than the GAN baseline (with the exception of SP4 where the mean fitness of NSLC-ViT is 2.9% lower than GAN-BSL). Overall, NSLC-HSV seems more stable in performance, reaching on average 1.5% higher mean fitness than GAN-BSL. By comparison, NSLC-ViT has more fluctuations between prompts and reaches an average increase

[5] https://github.com/lucidrains/big-sleep



(a) Mean population fitness

(b) Mean HSV diversity

(c) Mean ViT diversity

Figure 3: Progression of the performance metrics over GAN iterations. The iterations at which evolutionary NSLC cycles were performed are marked in red.

of 0.7% from the GAN-BSL mean fitness. The biggest increase in CLIP score is for SP3, where NSLC-HSV outperforms GAN-BSL by 3.9% in terms of mean fitness.

Figures 3b and 3c show how the mean diversity of the population fluctuates at different GAN iterations. Both image distance metrics are displayed, and the interim populations of all three methods (GAN-BSL, NSLC-ViT, NSLC-HSV) are parsed to derive these diversity values—even if they were not evolving towards that specific novelty measure. It is fairly surprising that for both image distance metrics the diversity increases during the first 20 GAN iterations. One would expect that the swift increase of the CLIP score (see Fig. 3a) during those early stages would come at the cost of diversity as the images are pushed towards a generic style imposed by the manifold. For both image distance metrics, the diversity for the GAN-BSL stays fairly stable after these first few iterations, or tends to drop. This is most pronounced in SP5 for both ViT diversity and HSV diversity; we hypothesise that the (literal) prompt itself pushes images that are fairly similar in colour (red and blue) and in terms of image classification. Regarding the NSLC variants, we observe the reverse behaviour compared to the mean fitness plots of Fig. 3a: diversity increases after each exploration cycle, at least for the distance metric targeted by novelty search. Interestingly, NSLC-HSV manages to increase both HSV diversity and ViT diversity, even if it evolves towards the former. On average, in each exploration cycle NSLC-ViT increases ViT diversity by 25% while NSLC-HSV increases ViT diversity by 10% (per prompt). NSLC-ViT however underperforms in terms of HSV diversity, with minor or no increases after each cycle. On the other hand, with NSLC-HSV we observe an average increase of 43% in HSV diversity after each cycle (per prompt). Since images produced by NSLC are more diverse but less fit, once GAN iterations re-start the diversity quickly drops as CLIP score increases. GAN iterations after NSLC tend to lead the population to a lower ViT diversity than the GAN baseline. This behaviour is surprising, especially considering that both NSLC variants manage to increase ViT diversity during the evolutionary cycles. Even more surprising is the fact that the GAN increases ViT diversity quite dramatically when dealing with random images (at 0 iterations), but this does not seem to be the case when NSLC produces noisy images at iterations 100, 200, 300, 400. After 600 iterations, the final images of NSLC-HSV have an average of 6.3% increase in HSV diversity compared to the GAN baseline but an average 11.5% decrease in ViT diversity, per prompt. The final images for NSLC-ViT however are less diverse for both ViT and HSV compared to the GAN baseline (by 5.8% and 13.7% respectively).

Indicative results

In order to better understand the process introduced in this paper, we show the most diverse images at different stages of the process. We use the HSV diversity and measure only the nearest-neighbour distance to choose the most diverse

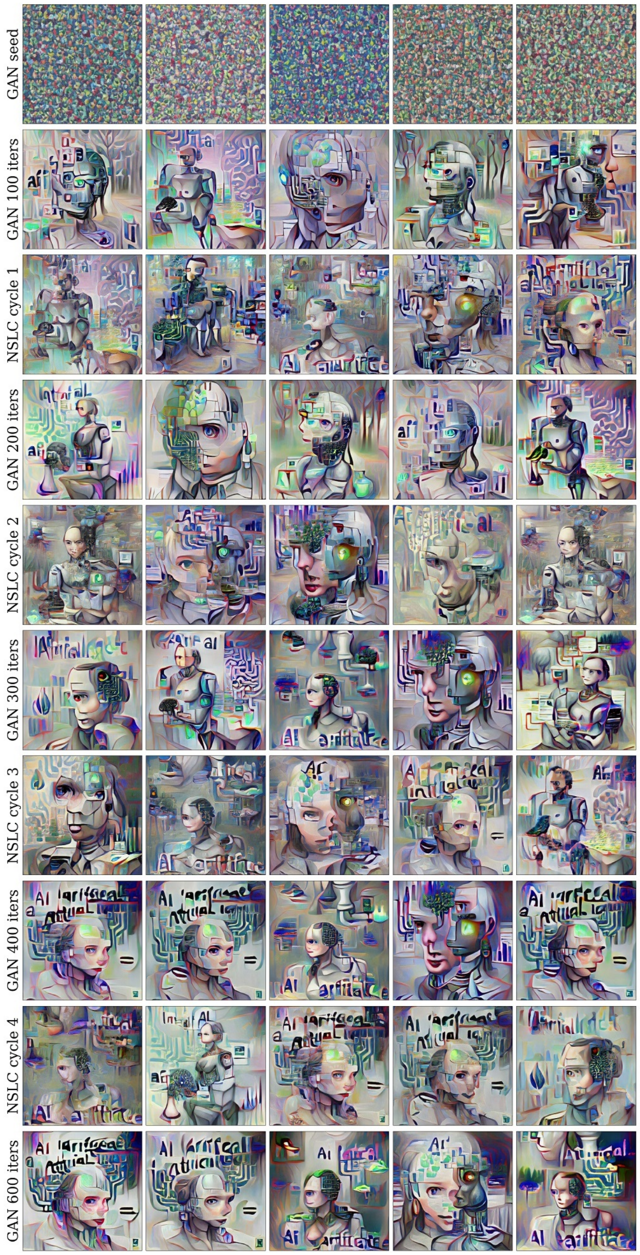


Figure 4: The progression of 5 individuals (chosen for their highest nearest-neighbour HSV diversity) per population at the end of each stage in the NSLC-HSV experiment for SP3.

individuals at that point. Since NSLC-HSV led to more diverse individuals while maintaining comparable quality to the GAN baseline, we show results of NSLC-HSV in Figure 4. For the purposes of brevity, we focus on SP3 since its final products have the highest increase in terms of CLIP score (3.9% above the GAN baseline) and a good increase in HSV diversity (7% above the GAN baseline).

It is evident that even after 100 GAN iterations, images are recognisable although their details are rough. At 100 GAN iterations, images are fairly diverse, while after the first exploration cycle new patterns are introduced (e.g. a human nose) but some images become more indistinguishable. These rough images are refined during the next GAN cycle, which results in similar-looking but crisper images. Similar rounds of exploration and refining add more details. Later NSLC cycles result in more recognisable, less noisy images. Notably, after 400 iterations the images start becoming more similar, and overarching patterns such as the introduction of the text “artificial” starts appearing in most images. At that stage, the last NSLC cycle does not quite manage to break these patterns and the final products at 600 iterations show more similarities than e.g. interim images at 300 iterations. We can assume that NSLC is more meaningful in early stages, and given enough time GANs will enforce their patterns even to initially novel images. Perhaps stopping the process earlier or intervening with NSLC in earlier stages (e.g. at 20 or 50 iterations rather than at 400) may better counter this drift towards dominant patterns.

Discussion

Our experiments investigated how quality-diversity evolutionary search applied in interim phases of a Generative Adversarial Network process can impact the creative process and products. Results show that seeding diversity in exploration cycles through NSLC can increase the diversity temporarily, but with a lesser impact in the long run as the GAN process re-asserts patterns in the corpus. While simplistic, HSV distance was shown to be better as a measure for the novelty score that guides evolution. However, observing the most diverse images in terms of HSV distance (see Fig. 4) the differences are not as obvious to a human. It is also worth noting that this study is the first to assess the diversity of a population of random initial seeds refined through the GAN iterative process; the final products were surprisingly more diverse than expected. As a general overview, NSLC manages to increase slightly the typicality (in terms of semantic prompts) of the final generated images; however, the small increase in diversity (and only for one visual similarity metric) compared to random seeds is perhaps underwhelming considering the computational overhead of multi-objective evolution over multiple cycles throughout the process. Despite these mixed results, the notion of diversifying products of AI Art has many interesting research directions beyond the experiments reported in this paper.

While this paper explored visual diversity under different perspectives (based on models trained on labelled data and based on simple visual metrics), there are many more ways. Other measures based on deep learning, such as the Learned Perceptual Image Patch Similarity (LPIPS) metric (Zhang et al. 2018) can be used both as a distance metric for novelty search or as a way to evaluate the existing products’ diversity. In our preliminary experiments using LPIPS for novelty score, however, the final products were not as diverse as those of the GAN baseline (in term of LPIPS). Given that HSV distance was surprisingly efficient as a novelty metric, other metrics of visual quality in the literature such as compressibility (Machado et al. 2015) could also be explored. It should be noted that in our preliminary experiments we also explored using the binary distance[6] between latent vectors (i.e. the genotype) as a measure of novelty, but the results were underwhelming.

Beyond the distance measures, other ways of performing changes on the image during evolution can be explored. While our preliminary experiments that used recombination between two parents' latent vectors resulted in less diverse final products, better operators for mutation and recombination could lead to more creative outcomes. A potential alternative to the current random mutation of the latent vector could be to use the intermediate representation used by the GAN, which consists of a tensor of real values, in order to provide a smoother gradient if mutation is based on Gaussian noise. The disadvantage to such an approach is an increase in computational time, since this intermediate representation is much larger than the latent vector used in our current work. Another alternative would be to apply mutations on the image itself, and then allow these to be decoded into a new latent vector (rather than the reverse, which is done in the current implementation). Changes to the image can be performed as filters applied to the entire image, similar to (Colton, Valstar, and Pantic 2008; Heath and Ventura 2016), as local changes in a portion of the image, or taking advantage of machine-learned models such as style transfer (Gatys, Ecker, and Bethge 2016).

Extensions of this work that go beyond applying NSLC on the images themselves could provide a more direct way to demonstrate the intentionality of the computational creator. OpenAI's CLIP already offers a human understandable (Colton 2008) goal in the form of the semantic prompt. Allowing the computational creator to adapt the semantic prompt itself (e.g. by applying latent variable evolution on the semantic prompt, rather than on the image) could lead to more visually diverse images and—more importantly—to a creative process where the computational creator could change its goal and explain towards which direction it is changing (and why, presuming some objective or distance criterion). More ambitious goals in this vein could include both image adjustments (through evolution) and a corresponding change in the best semantic prompt that matches these image adjustments. Finally, the refinement could come in the form of additions to the semantic prompt, such as maximising or minimising cosine similarity with keywords (e.g. "photorealistic") or with intended emotional outcomes from the audience (Galanos, Liapis, and Yannakakis 2021) that are added during exploration cycles. Further work in this direction could involve a human audience assessing diversity of the resulting images, thereby highlighting how the metrics match (or not) human perception and aesthetics.

Conclusion

In this work, we highlighted how what is considered today "AI Art" (McCormack, Gifford, and Hutchings 2019) largely ignores any creative dimensions except typicality (Ritchie 2007). We explored ways of injecting novelty both in the final products and in the process of a generative adversarial network, by interspersing cycles of artificial evolution that targets both typicality and novelty as objectives. Applying several cycles of exploration between cycles of iterative refinement, we investigated how image generation driven by state-of-the-art image-language mappings can lead to more diverse outcomes. This first experiment has shown that Novelty Search with Local Competition can lead to more visually diverse results, but also highlighted that evolution applied on the code book led to more noisy interim results which forced GAN refinements to overcompensate in terms of conformity. Many extensions to the general concept of cycles of evolutionary exploration and backpropagation-based refinement in different aspects of the AI Art process (e.g. on the image level or the prompt level) can allow for a more direct and more explainable creative process.

Acknowledgements

This project has received funding from the European Union's Horizon 2020 programme under grant agreement No 951911.

Author Contributions

Marvin Zammit prepared and carried out the reported experiments. Marvin Zammit and Antonios Liapis jointly analysed the resulting data. Marvin Zammit and Antonios Liapis each contributed to the writing in the various sections of the paper. Antonios Liapis and Georgios N. Yannakakis advised on the research direction and the text, and oversaw the implementation, analysis, and authoring process.

References

Blank, J., and Deb, K. 2020. Pymoo: Multi-objective optimization in python. *IEEE Access* 8:89497–89509.

Bontrager, P.; Roy, A.; Togelius, J.; Memon, N.; and Ross, A. 2018. Deepmasterprints: Generating masterprints for dictionary attacks via latent variable evolution. In *Proceedings of the IEEE International Conference on Biometrics Theory, Applications and Systems*.

Brock, A.; Donahue, J.; and Simonyan, K. 2018. Large scale GAN training for high fidelity natural image synthesis. *ArXiv preprint* abs/1809.11096.

Cohen, P. 2017. Harold Cohen and AARON. *AI Magazine* 37(4):63–66.

Colton, S.; Charnley, J.; and Pease, A. 2011. Computational Creativity Theory: The FACE and IDEA models. In *Proceedings of the International Conference on Computational Creativity*.

Colton, S.; Valstar, M. F.; and Pantic, M. 2008. Emotionally aware automated portrait painting. In *Proceedings of the International Conference on Digital Interactive Media in Entertainment and Arts*, 304–311.

Colton, S. 2008. Creativity versus the perception of creativity in computational systems. In *Proceedings of the AAAI Spring Symposium: Creative Intelligent Systems*.

Colton, S. 2012. *The Painting Fool: Stories from Building an Automated Painter*. Springer, Berlin, Heidelberg. 3–38.

[6] We measure binary distance as the number of items in the two images' latent vectors that were not identical at the same position.

De Jong, K.; Fogel, D.; and Schwefel, H.-P. 1997. A history of evolutionary computation. In Bäck, T.; Fogel, D. B.; and Michalewicz, Z., eds., *Handbook of Evolutionary Computation*. IOP Publishing Ltd.

Deb, K.; Pratap, A.; Agarwal, S.; and Meyarivan, T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2):182–197.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *IEEE Conf. on computer vision and pattern recognition*, 248–255.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the international Conference on Learning Representations*.

Esmaeilpour, S.; Liu, B.; Robertson, E.; and Shu, L. 2021. Zero-shot open set detection by extending CLIP. *ArXiv preprint* abs/2109.02748.

Esser, P.; Rombach, R.; and Ommer, B. 2021. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12868–12878. Los Alamitos, CA, USA: IEEE Computer Society.

Gal, R.; Patashnik, O.; Maron, H.; Chechik, G.; and Cohen-Or, D. 2021. StyleGAN-NADA: CLIP-guided domain adaptation of image generators. *ArXiv preprint* abs/2108.00946.

Galanos, T.; Liapis, A.; and Yannakakis, G. N. 2021. AffectGAN: Affect-based generative art driven by semantics. In *Proceedings of the ACII Workshop on What's Next in Affect Modeling?*

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2016. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2414–2423.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial networks. *Advances in neural information processing systems* 27.

He, T.; Zhang, Z.; Zhang, H.; Zhang, Z.; Xie, J.; and Li, M. 2019. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 558–567.

Heath, D., and Ventura, D. 2016. Creating images by learning image semantics using vector space models. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Jordanous, A. 2016. Four PPPPerspectives on computational creativity in theory and in practice. *Connection Science* 28(2):194–216.

Karras, T.; Laine, S.; and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4401–4410.

Kim, G., and Ye, J. C. 2021. DiffusionCLIP: text-guided image manipulation using diffusion models. *ArXiv preprint* abs/2110.02711.

Ledig, C.; Theis, L.; Huszar, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; and Shi, W. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition)*. IEEE.

Lehman, J., and Stanley, K. O. 2008. Exploiting open-endedness to solve problems through the search for novelty. In *Proceedings of the Conference on Artificial Life*.

Lehman, J., and Stanley, K. O. 2011. Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, 211–218.

Lehman, J., and Stanley, K. O. 2012. Beyond open-endedness: Quantifying impressiveness. In *Proceedings of the Thirteenth International Conference on Artificial Life*.

Liapis, A.; Martínez, H. P.; Togelius, J.; and Yannakakis, G. N. 2013. Transforming exploratory creativity with DeLeNoX. In *Proceedings of the Fourth International Conference on Computational Creativity*, 56–63.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2013. Sentient world: Human-based procedural cartography. In *Proceedings of Evolutionary and Biologically Inspired Music, Sound, Art and Design*, volume 7834, LNCS. Springer. 180–191.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2015. Constrained novelty search: A study on game content generation. *Evolutionary Computation* 23(1):101–129.

Liapis, A. 2016. Exploring the visual styles of arcade game assets. In *Proceedings of Evolutionary and Biologically Inspired Music, Sound, Art and Design*. Springer.

Lindemeier, T.; Metzner, J.; Pollak, L.; and Deussen, O. 2015. Hardware-based non-photorealistic rendering using a painting robot. *Computer graphics forum* 34(2):311–323.

Machado, P., and Cardoso, A. 2002. All the truth about NEvAr. *Applied Intelligence* 16(2):101–118.

Machado, P.; Romero, J.; Nadal, M.; Santos-del Riego, A.; Correia, J.; and Carballal, A. 2015. Computerized measures of visual complexity. *Acta Psychologica* 160:43–57.

McCormack, J.; Gifford, T.; and Hutchings, P. 2019. Autonomy, authenticity, authorship and intention in computer generated art. In *Proceedings of the international conference on Computational Intelligence in Music, Sound, Art and Design*.

Mokady, R.; Hertz, A.; and Bermano, A. H. 2021. ClipCap: CLIP prefix for image captioning. *ArXiv preprint* abs/2111.09734.

Pugh, J. K.; Soros, L. B.; and Stanley, K. O. 2016. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI* 3:40.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.;

et al. 2021. Learning transferable visual models from natural language supervision. *ArXiv preprint* abs/2103.00020.

Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; and Sutskever, I. 2021. Zero-shot text-to-image generation. *ArXiv preprint* abs/2102.12092.

Reed, S. E.; Akata, Z.; Mohan, S.; Tenka, S.; Schiele, B.; and Lee, H. 2016. Learning what and where to draw. *Advances in neural information processing systems* 29:217–225.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17:76–99.

Stanley, K. O., and Miikkulainen, R. 2002. Evolving neural networks through augmenting topologies. *Evolutionary Computation* 10:99–127.

Tan, W. R.; Chan, C. S.; Aguirre, H. E.; and Tanaka, K. 2016. Ceci n'est pas une pipe: A deep convolutional network for fine-art paintings classification. In *Proceedings of the IEEE International Conference on Image Processing*, 3703–3707.

Van den Oord, A.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *ArXiv preprint* abs/1807.03748.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Proceedings of the Conference on neural information processing systems*, 5998–6008.

Ventura, D. 2016. Mere generation: Essential barometer or dated concept? In *Proceedings of the International Conference on Computational Creativity*.

Wu, B.; Xu, C.; Dai, X.; Wan, A.; Zhang, P.; Tomizuka, M.; Keutzer, K.; and Vajda, P. 2020. Visual transformers: Token-based image representation and processing for computer vision. *ArXiv preprint* abs/2006.03677.

Zhang, H.; Xu, T.; Li, H.; Zhang, S.; Wang, X.; Huang, X.; and Metaxas, D. N. 2017. StackGAN: text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, 5907–5915.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–595.

Is style reproduction a computational creativity task?

Daniel G. Brown
Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
dan.brown@uwaterloo.ca

Anna Jordanous
School of Computing
University of Kent
Canterbury, Kent, UK
a.k.jordanous@kent.ac.uk

Abstract

Is style reproduction a valid computational creativity task? Does producing output 'in the style of' an existing creator contribute to computational creativity research? Where is the creativity in imitation or replication of an existing style, and where does style reproduction fall into what has been criticised as 'pastiche' rather than credible creative activity? This paper tackles these debates, which have been under-addressed in computational creativity literature. We review the presentaiton of past work in style reproduction, and consider the fit of such work into evolving definitions of computational creativity research. As part of this, we consider style reproduction itself as a creative task, both within and outside computational forms. We discuss various points of interest that emerge in the analysis, such as control in the creative process, intentionality and effort. Our work gives a more objective understanding of the level of creativity present in style generation, and specifically what value it brings to computational creativity research.

Introduction

Recently, there has been a striking increase in use of so-called "creative AI" systems. This rise has been particularly noticeable in two areas with low barrier to entry: text-generation systems like OpenAI's GPT family of transformers (Radford et al. 2019), and image-generation systems with generative adversarial networks (GANs) (Goodfellow et al. 2014), notably those inspired or derived from StyleGAN (Karras, Laine, and Aila 2018) and Creative Adversarial Networks (CANs) (Elgammal et al. 2017). In the former case, one can use a special corpus to fine-tune a general-purpose transformer to alter the parameterization of the neural network and enforce that the vocabulary and sentence style of a new text sample will be in similar style to training samples. In the latter case, one can use a collection of images of a variety of styles, and the neural network will generate new images intended to differ from all of those styles.

Creating and training new AI systems that generate new artifacts in a manner influenced by distinctive aspects of an existing creator, or *"in the style of"* that creator, is an exciting development, and it opens many areas of enquiry. For example, these new systems cannot merely commit plagiarism ("[t]he action or practice of taking someone else's work, idea, etc., and passing it off as one's own" (OED 2022)). We must ensure ethical use of corpora that may be of deceased authors on the one hand, or subject to copyright restrictions on the other hand (Brown, Byl, and Grossman 2021). Pease and Colton (2011) warn us off 'pastiche' (style imitation) to avoid compromising innovation and imagination. And focusing on older styles leaves computational art systems unprepared to respond to contemporary events.

But are these systems, and other systems that generate work "in the style of" their training data sets, computationally creative? How should the field of computational creativity respond to and integrate these new systems into our existing theories? Or do "in the style of" systems fall into a category below that of creative systems that are not merely replicating styles, but developing new ones? Here, we investigate this question by examining recent papers describing "in the style of" systems, both from inside the ICCC community and outside, and use existing theories of computational creativity to see which desiderata of those theories are and are not found in those papers.

Our overall conclusions are mixed. Style-reproduction systems can be computationally creative, however many fail to satisfy the goals of creativity theories, or only identify a system as creative due to human decisions. Our existing theories may need to be updated due to the ease of training standard models (like StyleGAN or fine-tuned GPT models) to emulate styles. In particular, one of Ventura's "lines in the sand" (criteria for creative systems) is that the system has a form of knowledge representation (Ventura 2016). But if all that is used is a standardized general model and fine-tuning procedure for a corpus scraped from a website, has the system meaningfully crossed Ventura's "line in the sand?"

The consequence of these general-purpose generative systems may be another round of the artificial intelligence "moving of the goalposts" that has happened repeatedly over the past several decades, moving various tasks such as photo retouching from one where detailed study time spent learning the practice could move one's photography to being "of new importance, and call[ing] forth words of approval" (Vicente 1904) to tasks largely done by a computer. Perhaps even "computationally creative" work requires substantial human labour to construct the system, forcing us back to focus on the human component of computationally creative systems in assessing whether they can be deemed "creative".

Style reproduction and human creativity

Style reproduction is the attempt to create novel creative works that are in the same genre and have stylistic elements in common with the creations of existing creators. In this paper, we are focusing on the emulation of the style of individual, specific creators: creating motets in the style of Palestrina, not just in that of the Italian Renaissance, for example, or weaving textiles similar to those of a specific fabric artist, not just those from a more general time and place; in practice, the lines between these tasks can be blurry.

When do humans do it, and is it creative?

Human beings reproduce style under many different circumstances. Many of these scenarios are educational: students may learn to write counterpoint in Bach's style as a school exercise in understanding Baroque harmony (Benjamin 1986), or they may create mock-Shakespearean sonnets to learn to write poetry (The Folger Library 2022). Even when they are not specifically commanded to duplicate an existing style, that may be the clear intent, as when they are exposed to still life paintings by a well-known painter and then asked to make a still-life of their own. These training tasks are not necessarily meant to create high-quality work (though, presumably, in some cases they do), and as the students are not experts in the work of the artist being emulated, the likelihood that the work would be particularly novel or reproduce the style well is also fairly low.

Experts also may reproduce styles as an homage. A hip hop example comes in a rap verse made by Bone Thugs-N-Harmony, when they reproduced the style of Notorious B.I.G. in a verse in the song “Notorious Thugs”, and vice versa; Biggie's verse in the style of the Bone Thugs helped make other prominent rappers take them more seriously (Findlay 2020). Poets emulate the style of their colleagues, particularly when writing odes to those colleagues. In other fields, style reproduction can allow established experts to learn more about the creative space: chess masters might attempt to play “in the style of” another player as a way of incorporating that player's ideas into their own play. In these cases, the expertise of the creator allows for high-quality novel work (within the scope of the copied style).

Another context in which creators create “in the style of” another creator can be in the visual arts, where an artist may make large-scale works requiring labour from a many participants. A muralist, for example, might plan a new large mural and then hire multiple artists to fill in the space devoted to the mural, all operating in a consistent style defined by the muralist. Similar circumstances may occur when artists work in a studio that builds smaller-scale art for sale that reproduces a primary creator's own work. Here, the creativity largely belongs to the primary conceptual creator, and the other hands on the project largely support that creator.

Another reason to duplicate the creator's style is to extend that author's oeuvre, particularly if it comes with a built-in audience. This has been done in “official” contexts, as with the dozens of “Oz” books written after L. Frank Baum's death in 1919 (Updike 2000). Similar, but related, is the creation of fan fiction or fan art, when fans build new works based on a beloved setting (Thomas 2011). Some fan art or fan fiction is “in the style of,” in the sense that it truly attempts to reproduce the original creator's vision; others can be “inspired by,” in the sense that it uses characters or situations from an original creator and adapts them to new circumstances the original author did not use. In both cases, quality can vary widely: much fan fiction is sloppy and a transparent facsimile of the original, but in some cases, fans do build successful creative works. For example, “Fifty Shades of Gray” was originally developed as “Twilight” fan fiction (CBC 2015), and the Archive of Our Own (AO3), hosts a number of extremely popular fan fiction stories, and even received a Hugo Award in 2019 for its cultural significance (Romano 2019). A further example of this kind of style transfer comes when a collective pseudonym is used for a collection of different creators, as with the “Hardy Boys” children's literature series, ghostwritten by a variety of authors under the name Franklin W. Dixon (Tensley 2019).

And of course, humans reproduce style for more nefarious reasons,like copying the style of a successful artist to sell forgeries; this process may occur most notoriously in the visual art world (Chernick 2020), but also fake manuscripts can also be used to pretend a deceased author had written things that they had not (Stewart 2010). Successful forgers meticulously copy the oeuvre of the artist whose work they are copying (sometimes even reproducing artistic media and materials), meaning that the space for them to be imaginative is vastly reduced; while they may produce technically excellent copies of a style, they may not be very novel.

Is human style-reproduction creative? In the cases we have described, many examples are not very high in creativity. The restriction to copy a well-established style may assist students in learning how to use artistic media or language, but the overall likelihood they create high-quality work is low. Here, a measure of quality we have in mind is one of significant computational effort, for example as formalized in Mondol and Brown (2021a; 2021b). Depending on how much of a “paint-by-numbers” approach the copied style has, a skilled copyist might reproduce the style faithfully, but this might indicate the overall lack of scope for novelty and quality in the original creator's work, implying that it, itself, is not creative. There is a tension: if reproducing style is akin to use of a photocopier, then there is minimal scope for creativity, as there is no room for novelty. If the task is more open, as with some fan fiction writing, it allows space for the new creator to genuinely explore a creative (albeit constrained) space, and it can be creative.

To be more specific, every aspect of the Four P analysis of creativity (Producer, Product, Process, Press) (Jordanous 2016; Rhodes 1961) can support the decision of the extent to which the task of creating artefacts “in the style of” some selected style is (or is not) a creative task in a particular context. The Producer can be exploring her personal identity in building works inspired by a beloved creator whose works have moved her, or she might be just trying to make a quick buck. The Product may be an excellent recapturing of the reproduced style, or it can be a sloppily-produced pastiche easily recognizable as both terrible and a sloppy copy of the original style. The Process can involve detailed research into the history and background of the copied creator and their methods, and a careful and laborious re-enactment of their ideas, or can focus on easy ways to slap up something that has surface features in common. And those who experience the Product (the Press) may either see it as yet another in a long line of tacky examples of a sad effort to capitalize on a once-beloved creator, or may celebrate the opportunity to re-engage with an oeuvre with slightly different eyes.

"Is human style reproduction creative?", like so many questions in creativity research, has the answer "it depends." But "yes" is certainly a possibility.

When computers reproduce style

We now consider research papers about automatic style reproduction. This literature is sparse; sparser still is discussion of the creativity of the task itself. We analyse several works both from within and outside the ICCC community, and focus on desiderata and frameworks to analyze computational creativity research.

Existing literature: a quick summary

In computer graphics, particularly non-photorealistic rendering, understanding a painter's style well enough to mimic it comes up particularly with distinctive painters, like the TV painting artist Bob Ross (Kalaidjian 2007) or Eyvind Earle (Murphy 2015), who was most responsible for the moody imagery in Disney's "Sleeping Beauty". In these cases, researchers were mainly interested in technical issues of the artists' styles,. According to a member of this research community (Kaplan 2021), this is often the goal of such work, not to either assess the creativity of new creations or to engage with the question of the overall task.

Successfully reproducing style has been treated as a fitness test for evolutionary computation, particularly in visual art and music (e.g. (Blackwell and Bentley 2002; Uhde 2021)). Uhde (2021) defines artistic style transfer as generation of new artefacts with the style of one input example and the content of a second input example. Though Uhde acknowledges the difficulties in distinguishing style from content, style identification and preservation is key to Uhde's formalisations. Bentley (1999) has presented deviation from an original guiding style towards a distinct new style as a problem, rather than a benefit, as it diminishes the contributions of the artist whose work was used as a guide.

One ICCC example of style reproduction is the DeepTingle paper from 2017 (Khalifa, Barros, and Togelius 2017). This work attempts to reproduce the distinctive style of the alarmingly prolific gay erotica author Chuck Tingle, using LSTM networks to produce new sentences and stories. The paper does not engage with the question of whether authoring stories in this way is a creative task, and uses A/B tests to compare the texts generated by the LSTM (or by a Markov chain) to those by the original author, on the categories of grammatical correctness, coherence, and interestingness. The authors highlight the challenge in duplicating a complex, unique style; they do not question whether duplicating art created by a marginalized author is appropriate.

Another ICCC paper presents EMILY, a system to create poems in Emily Dickinson's distinctive style (Shihadeh and Ackerman 2020). EMILY uses Markov chains custom-trained to focus on elements of Dickinson's poems. The quality of poems generated are compared to those of Dickinson on standard metrics (such as typicality, imagery and emotionality); Dickinson's poems score better than the ones they derive. Other similar papers reconstruct poetry in the style of Bob Dylan (Barbieri et al. 2012), Dante (Zugarini, Melacci, and Maggini 2019), Shakespeare and Oscar Wilde (Tikhonov and Yamshchikov 2018).

In the space of visual art, more recent projects like StyleGAN (Karras, Laine, and Aila 2018) train neural models to produce art indistinguishable by an adversarial network from art created by a specific creator. These systems reproduce style alarmingly well. However, the best possible outcome for such a system would be for it to create artifacts identical to or very similar to those from the training data set: novelty is not a direct goal. For that matter, neither is value: if the training data were all cartoons scribbled by children in crayons,[1] recreating that style would be the goal. Knowledge is represented in these systems, but the complex way in which neural networks represent goals makes answering "why" questions almost impossible currently.

By contrast, the Creative Adversarial Networks of Elgammal *et al.* (2017) were designed to create artworks of high quality (having properties similar to a training set) and novelty (style distinguishable from all styles in a training set). They do the opposite of style mimicry: they use the inspiring set, pre-divided by style, as a measure of what to avoid.

As with any computationally creative system, style duplication algorithms can incorporate the input of human co-creators. In one case, Kerdreux, Thiry, and Kerdreux (2020) focus on using a computer as a tool in helping an artist transfer the style of one image to another. They argue that the style-transfer algorithm *is* creative, because it can create images that have "an aesthetic that can significantly differ from what a painter would do" (i.e. an aesthetic that has broadened out beyond the inspiring style). Their focus was evaluating the images created by the collaboration between the system and the human, and in particular how to assess the quality of the collaboration between them. Co-creativity emphasises the importance of human participants perceiving their computational partners as a creative collaborator contributing in their own right (Jordanous 2017). Similarly, Crnkovic-Friis and Crnkovic-Friis (2016) produce choreography "in the style of" (though probably in more general style than that of a single choreographer). Their focus is on the ability of their neural network system to collaborate with humans, highlighting: "how current results can be used as a practical tool for a working choreographer." Hence style duplication can complement co-creativity - and vice versa.

Themes and goals of a style duplication algorithm

A striking absence from the papers we have discussed, and others we have found, is the key question of whether the

[1] Our inspiration for choosing this example is the second author's pride in her daughter's highly creative drawings.

underlying task of style reproduction is properly seen as a creative task, and specifically, as a computational creativity task. Even for the small number that have been published in the ICCC community, the goal has been faithful re-interpretation of the base style, and on what kinds of constraints need to be added to a base creative system to make it compatible with a new author's style, as with EMILY's needing to be adapted to deal with Emily Dickinson's punctuation choice , or the DeepTingle system's reproduction of Chuck Tingle's unusual vocabulary and grammar choices.

The more recent development of general-purpose systems that can be fine-tuned to reproduce individual creators' work also envisions a breadth of style reproduction work that is only just now starting. Authors both in the academic space and those from the popular press are using systems that simplify the process of fine-tuning of methods like GANs and language transformers so that culture hackers and creators can play around with "in the style of" creations, rather than focusing on those details. Even still, these methods are not citing whether their underlying methods are creative.

And finally, a key theme is co-creativity: many of these systems envision creators using them in context of those creators' work, rather than just running the systems full-bore and not curating or editing the results. For example, when Melynk (2021) used StyleGAN to create knitting patterns, she did not just design knitting patterns in the style of Fair Isle knitting, she also knitted the patterns themselves, and briefly discussed changes to make them fit the style better and work better as physical objects. In general, we see a large number of these researchers using "in the style of" creators as collaborators in their production process.

Other desiderata for computational creativity

Here we engage with other models of computational creativity in light of recent works of systems that build "in the style of", to further our analysis of whether this task is a computationally creative task.

The ICCC community stamp of approval

First, perhaps, there is the obvious fact that many papers have been accepted to the International Conference on Computational Creativity. Some of these are on the margin of the specific task under consideration: the CAN paper of Elgammal *et al.* (Elgammal et al. 2017) tries to push away from known styles, for example, and the six-word stories papers of Spendlove and Ventura (2020) and of Zabriskie, Spendlove, and Ventura (2018) discuss specifically genre, rather than style. However, the porosity of the boundary between these two versions of "in the style of" may be a key finding of our paper. Firmly in the "in the style of" category, however, are EMILY and DeepTingle, described above.

Further, we note the existence of papers that *imply* the computational creativity of this task, while analyzing other properties of such systems. For example, the ICCC best paper by Ens and Pasquier (2018) uses complexity measures to identify which style (including which creator) matches a given creative object best, and Brown, Byl, and Grossman (2021) consider the Canadian legal status of collecting special-purpose corpora for fine tuning of language models. There appears to be a willingness to at least consider the 'in the style of' task as legitimate by ICCC researchers.

Do the authors present their systems as creative?

Surprisingly few authors in the papers we have studied do describe their work as creative. While many of the ICCC authors follow a familiar-to-ICCC pattern of justifying (or at least stating) that the systems they produce are creative, many ICCC authors shy away from describing the systems they are presenting as computational creativity.

For the non-ICCC works, descriptions of the work as creative are strikingly absent: as noted above, theses reproducing the styles of both Bob Ross (Kalaidjian 2007) and Eyvind Earle (Murphy 2015) simply do not engage with the question of creativity at all. A law review article (Gervais 2019) describing the question of copyright of AI-derived works, which does do some engaging with the question of style reproduction ultimately argues (in a fashion familiar to ICCC researchers) that creativity is a fundamentally human endeavour and thus impossible for computers to perform.

A sophisticated non-ICCC example of "in the style of", which focuses on reproducing the style of a community, are the contests by Sturm et al., who highlight the social and cultural aspects of producing good folk songs (Sturm and Ben-Tal 2021). These researchers focus strongly on questions of ownership and appropriation, and perform extremely detailed and thorough evaluations, but still have not spent much time on the creativity question, let alone the computational creativity question.

Definitions of computational creativity

We can compare the papers we read to specific definitions of computational creativity.

The current ACC definition (Association for Computational Creativity 2014) extends the field to include algorithmic understanding of human creativity and to include co-creativity. As such, discussions of co-creativity, as in Kerdreux, Thiry, and Kerdreux (2020), clearly fit. None of the papers we considered spent much time on illuminating human creativity; the non-photorealistic rendering ones, for example, focused on technical issues of simulation, not on the process by which the creators worked.

This leaves the more traditional question of computational creativity: is the system capable of human-level creativity? While there are various ways to express this concept (see Jordanous (2014) for explanation), this frame is consistent with both the previous ACC definition and the popular Final Frontiers definition by Colton and Wiggins (2012).

There is evidence that the authors of some systems do see their work as attempting a task that would be human-level creative: for example, the EMILY paper (Shihadeh and Ackerman 2020) compares its work to real Emily Dickinson poems, and the lovely paper on identifying and naming new constellations (Sewell, Christiansen, and Bodily 2020) includes the strong claim, "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". In some cases, the evaluation of a system asks humans to assess the output on scales meant to assess creativity, as well. Whether these systems succeed or not, their authors believe that assessing them on their creativity is appropriate.

Desiderata for computational creativity

Colton's tripod criteria Colton's "creative tripod" (2008) identifies key criteria he argues are necessary for a creative system: skill, appreciation and imagination. "In the style of" systems built upon existing general-purpose creators (like StyleGAN or GPT language models) essentially outsource their skill and imagination to other systems (or to a human co-creator); further, to the extent that they are "appreciative", it is largely that those systems' general-purpose fine-tuning methods allow parameterizations to be learned from diverse sources without care for what makes a particular style special. In many other systems, imagination seems to be lacking, or largely comes in from human co-creators.

By contrast, special-purpose systems, like the constellation-identification paper (Sewell, Christiansen, and Bodily 2020), are implicitly appreciative: designed to identify and recreate the interesting aspects of their domain.

Ventura's standards Ventura also identified standards for a computationally creative system in two papers: his "mere generation" paper and "how to build a CC system" papers (2016; 2017) require the possible creation of novelty and value, and argue for intentionality and knowledge representation as key ways to avoid "merely generating."

Style reproduction systems run into serious problems in this frame. Intentionality, of course, is uncertain for most of them: as we note below, these systems have little to no autonomy in most cases, and they only reproduce a certain style because they are programmed that way. But novelty is also a serious concern: as a system's space of operation is constrained by its code, it may not be able to generate anything truly unusual; for example, DeepTingle does not have the astonishing breadth of inspiration of the real Chuck Tingle; see also the discussion of cover bands below.

Knowledge representation is also a challenge: in particular for systems that fine tune general-purpose systems, it is a stretch to say that they represent knowledge about the style they reproduce. Certainly at the least, they offer no way for an external observer to query what form that knowledge takes. A system that attempts to highlight specific aspects of a style, as with the choreography system of Crnkovic-Friis and Crnkovic-Friis (2016) (even if the details are hidden inside neural network parameters) may have more legitimate claims to represent knowledge of that domain well.

FACE model The FACE model (Colton, Charnley, and Pease 2011) suggests four different criteria that creative systems could include, each of which can be subdivided into two forms, ^{g} and ^{p}. To test for a FACE criterion, we ask if the system can generate framing information, aesthetic measures, concepts for how they operate and examples/expressions of those concepts (^{g}form), and if they can generate methods for generating each of the above (^{p}form).

No systems exhibited abilities to generate framing information (natural language textual descriptions that describe the processes employed by the system). However this is typical given the low occurrence of computational creativity systems with framing information included, especially outside the FACE model team; so we do not treat the absence of "framing" as indicative of the system not being creative.

Another similar observation which did deviate somewhat from general computational creativity research was that the freedom to be able to generate new methods for generation (the ^{p} form of the criteria) was absent in all examples analyzed. While such a capacity is uncommon in many computational creativity systems presented, it has been explored to a greater extent than systems using framing information, either as actual work presented or as potential for the future. However none of the style reproduction papers analyzed highlighted any value in systems gaining this 'meta-generative' ability, to generate generative methods themselves. Indeed, the generative process was highly controlled in many of the papers examined: in the EMILY system, construction of the model was done heavily supervised by known domain knowledge, rather than the system being allowed to find the style itself. In the DeepTingle system, the researchers themselves placed focus on the style's unique vocabulary and syntax as the key items to be replicated.

The third point of interest arising from the FACE model analysis was in looking at how systems had aesthetic measures. Where systems did, the measure was often tightly coupled to the measure of how well the output fit previous examples, with little in the way of other measures being permitted. In other words, style generation was seen as the overriding aesthetic determiner, with little room for other aesthetic choices to be allowed within the system processes.

How does the work interact with the Four Ps?

A convenient framework for understanding creativity, and computational creativity, is Rhodes's Four Ps (Person/Producer, Process, Product, Press) (Rhodes 1961), adapted to the computational creativity domain by Jordanous (2016); the recent tutorial on evaluation by Lamb, Brown, and Clarke (2018) also uses this as a scaffolding.

None of the papers we explored focused on the creativity of the Producer (when it was a computer); some did discuss the creativity of the human whose style was being emulated. Similarly, little is said in these papers about the Press (which corresponds to the social millieu in which a creation finds itself), except for measure of significance of the style being duplicated. (One delightful exception is the one-pot seasonings presented at ICCC by Fu et al. (2019): their product went to market, and their research made it clear that one goal of the product was, in fact, commercial success!)

Instead, unsurprisingly, most analysis in these papers hangs on the Product or Process characterizations. For example, Kazakçi, Cherti, and Kégl (2016) concern themselves with details of good generative Process. The StyleGAN and CAN papers (Karras, Laine, and Aila 2018; Elgammal et al. 2017) go into great detail about the underlying neural networks algorithms and objectives in their work. The authors of EMILY explore why custom generation of language models (in their case, Markov chains) is more apropos than using off-the-shelf models (Shihadeh and Ackerman 2020). And most authors describe various ways in which they evaluate the quality of their results by presenting those Products to humans or algorithms for judgment.

Still, if an author frames their work on one or more of the four Ps, this does not fundamentally resolve whether an individual project, or the overall style-reproduction idea, is creative, and a valid computational creativity pursuit.

And some outliers

We also note some outliers that we found in our study, which may highlight why this overall task is tough to place.

At ICCC'19 Pebryani and Kleiss (2019) described a co-creative system assisting Indigenous Balinese creators in producing culturally significant complex textile weaving patterns; here, the tool is as much a tool for training a new generation of designers as a creative system in its own right. The creators of the system focus on questions of process in their work, while emphasizing the ethnographic work in their research. When we asked an expert in Indonesian textiles about this work, he also highlighted the openness of Balinese designers to the use of technological innovations, as long as the textiles built in this manner were not used as important cultural artifacts (Sullivan 2021).

Also, some ICCC papers start with the acceptance of the importance of style transfer and use it as a primitive for further analysis. In addition to the CAEMSI paper (Ens and Pasquier 2018) and the Brown, Byl, and Grossman paper about language model corpora (2021), Kerdreux, Thiry, and Kerdreux (2020) use style transfer as a primitive in their artistic practice research, and Mondol and Brown (2021b) describe styles, their codification, and their reproduction as a task for algorithms to do in their algorithmic information theory model of several computational creativity primitives. The existence of these manuscripts argue in favour of style transfer as a computationally creative process implicitly: if the task is a sub-task of another computationally creative process, or creates other valid computationally-creative research areas by its sheer existence, then presumably, it is itself a valid computationally creative task.

Domain-general analysis of style reproduction

We have analyzed individual research contributions looking at style reproduction, across multiple creative domains. We now reflect on the overall requirements and properties of the task of style reproduction that we have seen repeatedly.

Style reproduction: highly-constrained creativity?

In discussions above of individual research contributions looking at style reproduction, we often see the creative systems operate in a more tightly constrained domain than we might usually expect for a creative system. To say this another way: the limits on acceptable output are more closely bounded, such that the set of possible outputs is smaller and more tightly controlled. Constraints can affect creativity (Sternberg and Kaufman 2010). In experiments on how constraints on output acceptability affected levels of creativity demonstrated by story generation systems, McKeown and Jordanous (2018) found "a sweet spot for maximal creativity closer to the less constrained end of the spectrum", but also that tighter constraints in their experiments afforded greater creativity than if the systems ran virtually unconstrained. In a more theoretical sense, Mondol and Brown (Mondol and Brown 2021b; 2021a) studied the extent to which setting up constraints on valid (or preferred) outputs can still allow for some domains to have a breadth of quality and novelty be displayed by creators.

In some of the systems reviewed above, we see the style reproduction task being implemented as output generation with additional stylistic constraints placed on the output, for example the punctuation-based, vocabulary-based or grammar-based restrictions placed on the output of the EMILY or DeepTingle systems (Shihadeh and Ackerman 2020; Khalifa, Barros, and Togelius 2017). It would seem, therefore, that it could be useful to consider treating style reproduction as a highly-constrained form of creativity.

Components of creativity

It is tractable to analyze the "in the style of" task itself via Jordanous's components of creativity (Jordanous and Keller 2016). We can break down creativity into these constituent parts for a more fine-grained understanding of the creativity inherent (and lacking) in the task of style reproduction.

Many of the creativity components are not affected by stylistic constraints for "in the style of" tasks, including *Active involvement and persistence*, *Dealing with uncertainty*, *General intellect* and *Spontaneity and subconscious process*. In other words, the above components are neither prioritized nor de-emphasized by the restrictions of fitting output to replicate or reproduce a particular style.

For other creativity components, the consideration of those components becomes more specific. *Domain competence* increases in importance, with the required competence being increasingly focused on a solid recognition of the definition and fit of the system output to stylistic expectations. *Generating results* is typically required from creative systems. In style reproduction, the generation of results is a necessity if the system is going to be deemed creative. *Social interaction and communication* gains an additional facet: the importance of output being socially relevant and acceptable, as examples of artifacts in the target style. It is not enough for those systems to generate artifacts that it deems to be stylistically relevant; they must be deemed acceptable by the wider community as reproductions of the target style. *Thinking and evaluation* takes on an additional required step; the evaluation must consider to what extent the target style is reproduced in the outputs. *Value* similarly gains an extra aspect: the extent to which the system outputs are stylistically accurate contributes to system value.

On the other hand, the importance of some of the creativity components becomes de-emphasized, or refocused, posing some really interesting challenges for the validity of style reproduction as a creative task. *Independence and freedom*, as we see in the analysis of style reproduction as a task with strong constraints, becomes much more limited. Style reproduction systems have some independence, but much less than a more general system. *Originality* at first consideration, seems to be severely compromised, even though it is widely recognized as one of the two critical parts of creativity (alongside value) (Runco and Jaeger 2012).

There is, however, still scope for originality or novelty within the task of style reproduction. Above we discussed the lack of creativity for a human performing tasks that are the creative equivalent of a photocopying task, yet allowed more attribution of creativity to a human who is performing style reproduction tasks in a way which there is still scope for some originality. This fits in with Boden's exploratory creativity (Boden 1992), such that the full conceptual space of possibilities is being explored, without changing the structure of the conceptual space. Originality is compromised in style reproduction, but still possible. The extent to which originality occurs within a style reproduction task appears correlated with the perception of the creativity of the entity performing that task. *Progression and development*, as with originality, is compromised to some extent; the system can explore the development of what it is doing, and progress from one state or set of outputs to another. The boundaries constraining such development and progression are, however, dictated and limited by the stylistic constraints more than is typical outside of style reproduction. *Variety, divergence, and experimentation* again can be thought of using Boden's exploratory creativity. The system can exhibit variety, and can diverge and experiment, though must remain within the conceptual space of the style being reproduced.

One component that poses an interesting challenge for this analysis is *Intention and emotional involvement*. This component can still be present in style reproduction systems, as a system can hold "intentions" (however implemented) to reproduce the intended style, and it can still use some kind of emotional modelling in its processes if that is applicable. However what it cannot do is express any intentions or desires to go beyond the stylistic constraints it has to operate in. What if, for example, a human musician who makes a living as part of a cover band (a band that reproduces the musical style and outputs of a recognized existing artist) decides to produce their own music, becoming emotionally invested in their new musical direction? If that is acceptable for a human musician, then what would it mean for a style reproduction system to change its intentions and want to explore new creative directions? Is this a flaw in the system or an exciting development for creativity? Or, arguably, both?

Even leaving behind the questions about what happens if a style reproduction system starts to deviate from the "in the style of" task it is designed to do, we have gained some useful insights from analyzing the creativity of the task of style reproduction using its constituent components. A surprising amount of room for creativity emerges. Creativity can still be demonstrated, it would appear, within the stylistic constraints that the system is operating in - as long as there is some room for originality and exploration. Certain aspects of creativity relating to value judgments increase in importance, demonstrating the challenges involved in building a system with the expertise to work in an existing style.

The use of Turing tests

We have repeatedly noticed the use of modified Turing tests, where the artifacts created by a computational system are compared by untrained humans to those created by the human creator whose style is being emulated ("can you identify whether this painting was created by a computer or by XXX?"). This phenomenon is in general frowned upon in computational creativity research: Pease and Colton (2011), in particular, have pointed out that building systems to pass this modified Turing test encourages pastiche and copying of the sorts of surface features that humans might notice, while not really engaging with the creative substance of a genre.

"In the style of" creation, however, offers a situation where perhaps these modified Turing tests are appropriate as an evaluation, at least of the question of whether or not the style has been copied. (Obviously, every genius has bad days: just knowing that a poem reads like an Emily Dickinson poem does not mean it reads like a *good* Dickinson poem!) Still, many of the systems themselves, particularly those based on GANs, are themselves trained to confuse an *internal* system into being unable to distinguish true examples of the targeted style from those created by the system.

The question of intentionality and autonomy

In the previous section, we explored a number of frameworks developed to identify the extent to which style duplication systems can be computationally creative. A key take-away message is that existing systems miss out on a few of the elements of these systems, but the most serious lapse is *intentionality*. That is, there is no obvious reason why style duplication systems do what they do, and minimal scope to engage with intention or the ability to consider multiple styles for suitability. By contrast, computationally creative systems that *have* engaged meaningfully with the question of intention have mostly done so by beginning with a representation of knowledge, and then allowing the system to choose which events to report, and with which response.

For example, Ventura (2019) shows how DARCI chooses when to make a painting, which elements to include in that painting, and how to represent them. Similarly, Colton (2012) explains how The Painting Fool can answer the question "why did you paint this?" by reference to news articles it has read. A bot that retells a daily news story in the style of a famous politician, for example, lacks this sense of creative autonomy (it must always make a story) and lacks the intentionality needed to best represent the story. If, instead of following a single style, a creative system were able to choose an apropos style, based on the events or mood being conveyed, such a system might be better able to claim the mantle of autonomy and intentionality, at least at the level that existing systems that emphasize these features do.

Co-creative systems and intentionality

Multiple frameworks stress autonomy and intention as key elements of a creative system. This may, in fact, be a red herring. Perhaps we insist on these elements as we subconsciously seek a difference between humans and computers. Since computers are (perhaps with some layers of indirection) only programmed because of human intentions, we see a key concern that motivation must come from somewhere.

In theory, a co-creative system that allows a human creator to consider many different authors' styles might allow them the entertaining task of responding to one day's news with a movie script in *film noir* style, and the next day's news with the text of a Shakespearean sonnet. In this sense, the human task (that of intentionality and autonomy in choosing subject and style) and the automated task (that of representing an event or subject in that style) can be handled by each actor effectively. For that matter, the automated system might attempt to represent the event in multiple styles and leave it up to a human participant to be part of the process of choosing which style works best for a situation.

Does labouring matter?

One clear reason to develop style reproduction algorithms is to change the role of the human in the process: instead of doing the labour of figuring out which sentences of a creator's oeuvre might be apropos a specific inspiring event, or figuring out which cadence would properly represent a composer's work at the culmination of a piece, the human being can cast that task to the style reproduction algorithm. In particular, at this point, near-novices can build almost any "in the style of" model for English texts with relatively little work using existing GPT-2 worksheets written in Google Collab; one just must supply the text upon which the model must be fine-tuned (Woolf 2019). This has caused popular blogs like "AI Weirdness" to present silly examples of GPT-2's creations of British snacks, Halloween costumes and more. (These humorous weirdnesses happen in part because of overtraining due to the tiny fine-tuning data sets.)

We cannot shake the belief that these general-purpose fine-tuned generators really do change the level of creativity involved across the board. If one day, we build Shakespearean sonnets, the next day, we build odes in the style of Keats, and the following day, we build Imagist poems in the style of William Carlos Williams, it feels like the labour that has typified previous researchers and creators, painstakingly trying to account for the punctuation styles or vocabularies of existing authors, has vanished into the ether. We could even, in theory, write this paper paragraph-by-paragraph, translating each paragraph into a different creator's style. (We note that we have not done this.)

Moving the goalposts

However, the situation with other activities in which humans engage is that we have often down-graded the creativity of certain tasks after computers (and AI systems in particular) have gotten good at them. Some tasks are "still" typically considered creative, despite the assistance their computer collaborators give to humans. For example, crossword puzzle creators can access word lists (and even common clues) as they develop their puzzles, and it has been possible to fully generate such puzzles for many years (Rigutini et al. 2008), but the task of creating crosswords is still seen as creative. Similarly, comic book artists need not hand-shade their panels anymore. But some word puzzles may in fact be less creative (for solvers and designers alike) once their underlying algorithmic nature is identified. Similarly, some strategy games, like checkers, have been fully solved (in the sense that any player facing an optimal computer player will at best tie the computer player) (Schaeffer et al. 2007); does this mean that good game play was never creative? Does it mean it is no longer creative?

We believe these questions have been less addressed in the computational creativity literature than they should be; in particular, certain domains are so constrained by the "in the style of" constraint that they feel a bit automatic to enact. If the supply of good-quality haiku in the style of a single producer that respond to a single prompt is small, and the process of creating them is very standardized, then it would be unsurprising to see the ostensibly creative task get rounded down to being not-very creative. How much is our field participating in this general process of "rounding down" the creativity of tasks?

Conclusion

We think the answer to the question of our title is the unsatisfying answer, "sometimes". Arguing in favour of style reproduction being a computational creativity task: style reproduction requires the agent to produce novel and valuable work in a highly constrained space of valid possibilities, and properly imitating the style of a famous creator requires skill and appreciation. Building good paintings in Salvador Dalí's style is no different than building good Surrealist paintings.

Many of the systems we consider work hard to reproduce important features of the underlying style; others exploit general-purpose systems that can be adapted to discover these features. The systems often carry an underlying concept with them, and incorporate both aesthetics and evaluation into their internal processes; in many cases, this comes for free from the general-purpose systems upon which they are created. And, as is often true with current computationally creative systems, these systems routinely collaborate with human co-creators; if in these scenarios, the human finds the computer to be a valuable partner, that is strong evidence for the idea that the systems are computationally creative, and so is the task.

Arguing against the claim that computational style reproduction is computationally creative is the routineness and triviality of the adaptation to new styles: if all that is needed to turn GPT-2 from a Hemingway story generator to a Keats poem generator is to change the fine-tuning training data, then it might be hard to say that this task is worthy of the name "creative"; in particular, saying there is a true concept being carried by the general-purpose system through the generation process may be impossible. We also argue that the key goals of intention, autonomy and motivation are especially weak in the case of reproduction "in the style of", unless the answer is actually to be found in the mind of a human co-creator (or in the case of systems not yet built, in their own intentional decision of *which* style to reproduce.

Ultimately, "in the style of" creation is, perhaps, just a heavily-constrained version of any other computationally creative task, with reduced (but still present) scope for novelty. We hope that future researchers will look on it with an eye for all of the issues we have discussed in this paper, and will examine whether their solutions are computationally creative, or if they are just routine turning of the crank.

Acknowledgments

The work of D.G.B. is supported by an NSERC Discovery Grant. We appreciate useful conversations with Piotr Sawicki, Marek Grześ and Max Peeperkorn.

Author contributions

This work was done through close collaboration between the two authors.

References

Association for Computational Creativity. 2014. Computational Creativity.

Barbieri, G.; Pachet, F.; Roy, P.; and Esposti, M. D. 2012. Markov constraints for generating lyrics with style. In *Proceedings of the 20th European Conference on Artificial Intelligence*, ECAI'12, 115–120. NLD: IOS Press.

Benjamin, T. 1986. *Counterpoint in the style of J.S. Bach*. New York: Schirmer Books.

Bentley, P. 1999. Aspects of evolutionary design by computers. In Roy, R.; Furuhashi, T.; and Chawdhry, P. K., eds., *Advances in Soft Computing*, 99–118. London: Springer London.

Blackwell, T., and Bentley, P. 2002. Improvised music with swarms. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, volume 2, 1462–1467 vol.2.

Boden, M. 1992. *The Creative Mind*. London: Abacus.

Brown, D. G.; Byl, L.; and Grossman, M. R. 2021. Are machine learning corpora ”fair dealing” under Canadian law? In *ICCC*, 158–162.

CBC. 2015. Fifty Shades of Grey fan fiction devotees grapple with film's success.

Chernick, K. 2020. Art forger Han van Meegeren fooled the world into believing his fake Vermeers. A new film unpacks his bag of tricks. *ArtNet*.

Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier? In *ECAI*.

Colton, S.; Charnley, J. W.; and Pease, A. 2011. Computational creativity theory: The FACE and IDEA descriptive models. In *ICCC*, 90–95.

Colton, S. 2008. Creativity versus the perception of creativity in computational systems. In *Creative Intelligent Systems, 2008 AAAI Spring Symposium*, 14–20.

Colton, S. 2012. The Painting Fool: Stories from building an automated painter. *Computers and Creativity* 3–38.

Crnkovic-Friis, L., and Crnkovic-Friis, L. 2016. Generative choreography using deep learning. In *ICCC*, 272–277.

Elgammal, A. M.; Liu, B.; Elhoseiny, M.; and Mazzone, M. 2017. CAN: creative adversarial networks, generating "art" by learning about styles and deviating from style norms. In *ICCC*, 96–103.

Ens, J., and Pasquier, P. 2018. CAEMSI : A cross-domain analytic evaluation methodology for style imitation. In *ICCC*, 64–71.

Findlay, M. 2020. 2Pac and Notorious B.I.G. Made Classics With Bone Thugs-N-Harmony. *Hot New Hip Hop*.

Fu, J.; Goodwin, R.; Harris, C.; Lang, K.; Lougee, R. W.; McLane, C. J.; Maria, J.; Martin, J. J.; Segal, R.; and Yeshi, T. 2019. Computational-creativity enabled one pan seasonings for retail sale. In *ICCC*.

Gervais, D. 2019. The machine as author. *Iowa Law Review* 105:2053–2106.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial networks.

Jordanous, A., and Keller, B. 2016. Modelling creativity: Identifying key components through a corpus-based approach. *PLoS ONE* 11:e0162959.

Jordanous, A. 2014. What is computational creativity? *The Creativity Post*.

Jordanous, A. 2016. Four PPPPerspectives on computational creativity in theory and in practice. *Connection Science* 28(2):194–216.

Jordanous, A. 2017. Co-creativity and perceptions of computational agents in co-creativity. In *ICCC*.

Kalaidjian, A. 2007. Automated Landscape Painting in the Style of Bob Ross. Master's thesis, University of Waterloo.

Kaplan, C. 2021. Personal communication.

Karras, T.; Laine, S.; and Aila, T. 2018. A style-based generator architecture for generative adversarial networks. *CoRR* abs/1812.04948.

Kazakçi, A.; Cherti, M.; and Kégl, B. 2016. Digits that are not: Generating new types through deep neural nets. In *ICCC*, 188–196.

Kerdreux, T.; Thiry, L.; and Kerdreux, E. 2020. Interactive neural style transfer with artists. In *ICCC*, 437–444.

Khalifa, A.; Barros, G. A. B.; and Togelius, J. 2017. DeepTingle. In *ICCC*, 167–174.

Lamb, C.; Brown, D. G.; and Clarke, C. L. A. 2018. Evaluating computational creativity: An interdisciplinary tutorial. *ACM Comput. Surv.* 51(2).

McKeown, L., and Jordanous, A. 2018. An evaluation of the impact of constraints on the perceived creativity of narrative generating software. In *ICCC*, 16–23.

Melnyk, V. 2021. Punch card knitting pattern design in collaboration with GAN. In *ICCC*, 336–341.

Mondol, T., and Brown, D. G. 2021a. Computational creativity and aesthetics with algorithmic information theory. *Entropy* 23(12).

Mondol, T., and Brown, D. G. 2021b. Incorporating algorithmic information theory into fundamental concepts of computational creativity. In *ICCC*, 173–181.

Murphy, L. K. 2015. Procedural Generation and Rendering of Trees and Landscapes in the Style of Eyvind Earle. Master's thesis, Texas A & M University.

OED. 2022. Oxford English Dictionary Online.

Pease, A., and Colton, S. 2011. On impact and evaluation in computational creativity: a discussion of the Turing Test and an alternative proposal. In *AISB '11*, 15–22.

Pebryani, N. D., and Kleiss, M. C. 2019. Ethnocomputation: Culturally specific design application of geringsing textile patterns. In *CAAD Futures 2019*, 538–551.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners.

Rhodes, M. 1961. An analysis of creativity. *Phi Delta Kappan* 42:305–310.

Rigutini, L.; Diligenti, M.; Maggini, M.; and Gori, M. 2008. A fully automatic crossword generator. In *2008 Seventh International Conference on Machine Learning and Applications*, 362–367.

Romano, A. 2019. The Archive of Our Own just won a Hugo. That's huge for fanfiction. *Vox*.

Runco, M. A., and Jaeger, G. J. 2012. The standard definition of creativity. *Creativity Research Journal* 24(1):92–96.

Schaeffer, J.; Burch, N.; Björnsson, Y.; Kishimoto, A.; Müller, M.; Lake, R.; Lu, P.; and Sutphen, S. 2007. Checkers is solved. *Science* 317(5844):1518–1522.

Sewell, A.; Christiansen, A.; and Bodily, P. M. 2020. Creative constellation generation: A system description. In *ICCC*, 496–499.

Shihadeh, J., and Ackerman, M. 2020. EMILY: an Emily Dickinson machine. In *ICCC*, 243–246.

Spendlove, B., and Ventura, D. 2020. Creating six-word stories via inferred linguistic and semantic formats. In *ICCC*, 123–130.

Sternberg, R. J., and Kaufman, J. C. 2010. Constraints on creativity. *The Cambridge handbook of creativity* 467–482.

Stewart, D. 2010. To be ... or not: the greatest Shakespeare forgery. *Smithsonian*.

Sturm, B. L. T., and Ben-Tal, O. 2021. *Folk the Algorithms: (Mis)Applying Artificial Intelligence to Folk Music*. Cham: Springer International Publishing. 423–454.

Sullivan, C. 2021. Personal communication.

Tensley, B. 2019. The knotty nostalgia of the Hardy Boys series. *The Atlantic*.

The Folger Library. 2022. Write a sonnet.

Thomas, B. 2011. What is fanfiction and why are people saying such nice things about it?? *Storyworlds: A Journal of Narrative Studies* 3:1–24.

Tikhonov, A., and Yamshchikov, I. P. 2018. Sounds Wilde. Phonetically extended embeddings for author-stylized poetry generation. In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, 117–124. Brussels, Belgium: Association for Computational Linguistics.

Uhde, F. 2021. Applicability of convolutional neural network artistic style transfer algorithms. In *Artificial Intelligence and the Arts*. Cham: Springer. 61–81.

Updike, J. 2000. Oz is us. *The New Yorker*.

Ventura, D. 2016. Mere generation: Essential barometer or dated concept? In *ICCC*, 17–24.

Ventura, D. 2017. How to build a CC system. In *ICCC*, 253–260.

Ventura, D. 2019. *Autonomous Intentionality in Computationally Creative Systems*. Springer. 49–69.

Vicente, F. 1904. Artistic retouching, paper no. 1. *The Camera* 8:81–84.

Woolf, M. 2019. AITextGen. Google Collab notebook.

Zabriskie, N.; Spendlove, B.; and Ventura, D. 2018. An hbpl-based approach to the creation of six-word stories. In *ICCC*, 161–168.

Zugarini, A.; Melacci, S.; and Maggini, M. 2019. Neural poetry: Learning to generate poems using syllables. In Tetko, I. V.; Kůrková, V.; Karpov, P.; and Theis, F., eds., *ICANN 2019: Text and Time Series*, 313–325.

5. Music and applications

Neo-Riemannian Theory for Generative Film and Videogame Music

Sara Cardinale[1] and **Simon Colton**[1,2]
[1]School of Electronic Engineering and Computer Science, Queen Mary University of London, UK
[2]SensiLab, Faculty of Information Technology, Monash University, Australia

s.cardinale@qmul.ac.uk s.colton@qmul.ac.uk

Abstract

Music is an essential element of films and videogames, which strongly contributes towards an immersive experience, by establishing a setting and mood, enhancing the storyline, and developing characters. Automatically generating music for films and games has been explored in existing works, but there is still room for improvement in terms of musicality and adaptivity. Neo-Riemannian Theory (NRT) comprises an established set of techniques for analysing music which is triadic but not necessarily tonal, and has had much application to the study of emotional and situational arcs in film music. NRT has barely been used in a generative setting, and we introduce a rationalised version for this purpose, which we believe could have particular application to film and videogame music, where mood or emotion-based music is required. We suggest ways in which such a procedural NRT approach could be applied, and describe future directions for our own projects in this area.

Introduction

Music is central to the setting of ambiance, mood and emotional arcs in films and videogames, and often contributes to the portrayal of plot elements and character development. Generative music can be employed for offline film/game development and also in response to player actions and environment changes during live gameplay (Plut and Pasquier 2020). The adaptive nature of such generative music can often be beneficial, e.g., when experiencing music generated by the *Adaptive Music System* (AMS) (Hutchings and McCormack 2020), gamers reported "... an overall higher immersion and correlation of music with game-world concepts with the AMS than with the original game soundtracks ..." However, existing generative music systems for games can be said to lack somewhat in *musicality*, i.e., the ability to present music in a pleasing way, take sounds and arrange them in patterns and phrases using music theory concepts such as rhythm, harmony, dynamics, tone, articulation, form, musical continuity, and tempo to express thoughts and emotions. For instance, in (Hutchings and McCormack 2020), the authors state that: "listening to the music tracks, it becomes apparent that the overall music quality of the AMS is not that of a skilled composer ... musical quality could be enhanced through improvements to the composition techniques within the framework presented." Moreover, Liapis et al. state that there is not a substantial difference in the generation of game audio and music generation outside of games (Liapis, Yannakakis, and Togelius 2014).

In general, the systems used in industry are usually game specific, while systems made for academic research are too general (Plut and Pasquier 2020). For instance, the generative compositional system for *The Audience of Singular* videogame by Plut (2017) produces music which is not entirely adaptive and could be considered as an independent musical piece (Plut and Pasquier 2020). For melody generation, the system uses Markov Models to generate various possible music sentences or notes that are randomly selected from a scale. Such an approach does not follow organizational elements such as pitch proximity and late-phrase declination (Huron 2006) and can result in lack of musicality. Similar limitations can be found in generative systems for film music, such as in the *DeepScore* project (Savery and Weinberg 2020), where music for a film clip can be generated to fit temporal keyword cues, or to fit a visual analysis of the clip. Here, an evaluation of the system resulted in feedback that if the system used keywords instead of visual analysis of the media, the generated music was pleasant to listen to, but was often not in-sync with the on-screen actions, and the score was too simplistic and lacking emotion. A similar response was given for the system when using visual analysis, with additional feedback opining that transitions between scenes were too unharmonious.

We are developing a general generative system which can adapt to the requirements of a scene from a specific film or game to produce music with high musicality and appropriate support for the scene. To this end, we propose a procedural version of an established music analysis technique called Neo-Riemannian Theory (NRT). This has been used to successfully analyse emotional and situational arcs in film music, especially where *triadic* but not necessarily tonal music has been composed. With the term major/minor triadic chord (or trichord), we mean a chord of three notes with a tonic note, another note a major or minor third above the tonic, and a final note a fifth above the tonic. These can be presented in various permutations, or *inversions*. Our procedural version of NRT comprises a set of rewrite rules for minor and major trichords, and inherits a mapping from chord progressions to changes in emotional and situational elements in media such as films or videogames. Before describing NRT and procedural NRT, we first describe the roles that music plays in films and games. We end by describing related work and future directions for our work.

Proceedings of the 13th International Conference on Computational Creativity (ICCC'22)
ISBN: 978-989-54160-4-2

The Roles of Music in Films and Videogames

Films put demands on the music for their soundtracks that are quite different from those imposed on other genres (Lehman 2018), with videogames perhaps being the closest in this respect. In general, soundtracks must take on various roles to enhance and accompany the media's narrative. Copland delineates five purposes of film music: creating a convincing atmosphere of time and place; underlining psychological refinements (the unspoken thoughts of a character); serving as a kind of neutral background filler; building a sense of continuity; and underpinning the theatrical build-up of a scene, eventually rounding it off with a sense of finality. In film music, while the basic building blocks are the same as any other Western genre, how these are arranged is guided by the script, where the end goal of the music is making meaning (Lehman 2018), performing a narrative function opposed to solely being musically pleasing.

In order to further understand how soundtracks are composed to fit the video, Lehman (2018) sets out three Hollywood practices: (a) the soundtrack's active role in making meaning, (b) the tendency for film music to rely on immediate gestures for expressive impact where tonality is not necessary, via the practice of using small musical ideas chained together, repeated or changed to prevent the music from being overwhelming, by constantly giving the audience new musical information, and (c) the music's ability to make meaning via associations e.g., linking a feeling to a melody and being able to recall it through repetition.

Soundtracks for films and videogames share many functionalities. In both cases, timed musical cues and sound effects typically suggest a responsive, narrative-specific environment aimed at either immersing the viewer/player in the spectacle of storytelling or engaging them in the bodily emulation of problem solving in a narrative-based context. These principles give music its ability to create a compelling and entertaining emulation, as described specifically for game music in (Whalen 2004). Moreover, videogame music draws from various cinematic practices to help structure the game's narrative elements based on familiar dramatic conventions (Rod 2007).

Neo-Riemannian Theory for Musical Analysis

NRT originated in the work of David Lewin (1982), and was built on Hugo Riemann's work on interrelations of triads and systems. An example of this is harmonic dualism (negative harmony), which describes the inversional relationship between major and minor chords, with minor triads being considered "upside down" or mirrored versions of major triads (Rehding et al. 2003). NRT was a response to analytical problems created by chromatic music that is triadic but not necessarily tonal (*triadic chromaticism*) (Lewin 1982). Analytical models for diatonic music are not suitable to analyse chromatic chord progressions, as chromaticism makes use of notes foreign to 7-note modes or diatonic scales (Cohn 1998). Therefore, Neo-Riemannian Theory has been used as an analysis tool for chromatic chord progressions, including in film music, showing that tonality is not the only way to relate chords.

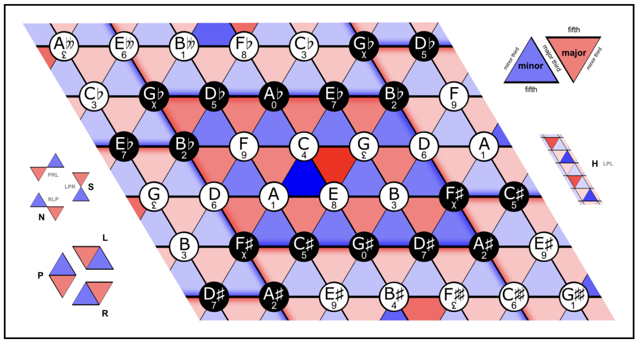


Figure 1: The Tonnetz, from en.wikipedia.org/wiki/Tonnetz.

NRT employs three Neo-Riemannian Operators (NROs) to describe the transition from one trichord to another. These are: *Parallel (P)* for pairs of triads that share an interval of a fifth, *Relative (R)* for triads that share a major third, and *Leading-tone Exchange (L)* for triads that share a minor third. When analysing music, the chord transitions which can be identified with these operators are dictated by the *Tonnezt* depicted in figure 1, which has been used since the 18th Century to describe chord progressions. To describe the P, L and R operations, we take a triangle on the main board of either red or blue colour and map it to the adjacent triangle, as per the key in the bottom left of figure 1. Red triangles represent major trichords and blue triangles represent minor trichords, and we see that P, L and R always transform major to minor chords and vice-versa.

Part of the analytical power of NRT lies in the ability to analyse chord transitions which are not captured directly as P, L or R transitions, but sequences thereof. For instance, if a transition of one triadic chord A to another B is captured as the NRO sequence LP, then this represents the fact that applying L to A, then applying P to the resulting chord will end with B. Lehman (2014) showed that such sequences of P, L and R transitions can model any possible relation between the major and minor triads. Subsequent additions to the theory introduced two inversional operators, *Slide (S)* which exchanges two triads that share a third (such as C major and C♯ Major), and *Nebenverwandt (N)* which transforms a major triad into its minor subdominant, and vice-versa, *(N')* a minor triad into its major dominant (Lehman 2014).

Triadic chromaticism appears in most film scores (Lehman 2014). An example is the soundtrack of the movie *A Beautiful Mind* composed by James Horner. Throughout the whole score, chromatic chord progressions, which can be analysed

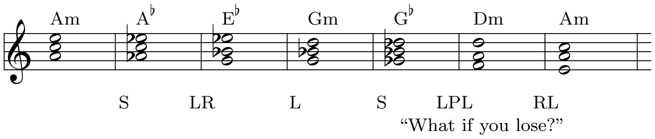


Figure 2: NRT analysis of an excerpt from the soundtrack to the film *Beautiful Mind*. The NRO chord transition sequence is given below the stave.

Table 1: Association of NRO sequences and emotional and/or situational scene elements (Lehman 2014).

NRO Sequence	Emotion/Situation
LP	Antagonism
L	Sorrow, loss
N	Romantic encounters
PRPR	Mortal threats, dangers
RL	Wonderment, success
NRL	Suspense and mystery
RLRL	Heroism (Lydian)
NR	Fantastical
S	Life and death

using NRT, are used to portray the genius of the protagonist. Figure 2 shows a small excerpt of the chord progression used in the cue *"Playing a Game of Go!"*, analysed using NRO. In the film at this point, the music highlights when the lead character loses a game of the board game Go and self doubt creeps in. This is an example of the music supporting an emotional event, which can be captured analytically using NRT. Lehman (2013) links such emotions and events to sequences of NRO operator applications, i.e., sequences of trichord changes. In this way, the mapping of the portrayal of emotional and/or situational aspects of a scene (such as the expression of genius, sorrow and wonderment) and the music accompanying it can be captured analytically. For the set of emotion/situational constructs captured in this way, see table 1. In the excerpt of figure 2, we see that S (life and death) operators, along with L (sorrow, loss) and LP (antagonism) sequences are employed. While the mapping of the music to the emotion/situations in table 1 is not exact, the analysis of the emotional arc of the music matches well the emotions of the film scene being acted out.

Sequences of NRT operators can generate transformations between chords that do not share any notes, and Lehman (2014) states that the transformational complexity and its connection to aural distance can convey feelings and meaning by considering the distance between the original chord and the destination chord. For instance, passages with simple LPR compounds can be associated with relaxation due to the closeness of chords. In contrast, complex combinations can be associated with tension. Furthermore, NRT's transformational approach leads to the possibility of writing tightly voice-lead passages of music. That is, as NROs transform chords into new ones that share common tones, the melodies produced from these operators could present linear progression, making the melodies easy to sing and hence probably more memorable. This is an important feature of soundtracks, which use melodies as story-telling devices (Whittall 2001).

For such film music, NRT is used to analyse chromatic chord progressions that would not be fully justifiable if analysed using a diatonic approach (Lehman 2014) as this type of music makes use of notes foreign to a mode or diatonic scale. Ensuring that the progression stays diatonic requires the use of specific combinations of PLR, therefore not utilizing NRT to its full potential. Furthermore, Lehman states that as NRT operates on pitch classes rather than diatonic notes, NROs disregard enharmonic spelling (e.g. C♯ and D♭ are the same note on a keyboard but the name of the note changes depending on the key signature). In film music analysis and likewise in film music composition, this allows one to focus on other musical features, such as associativity and meaning, that are more important to the genre, while avoiding claims that such chromatic progressions are due to some unjustifiable or irrational music theory.

Table 2: NRT Rewrite rules for major and minor chord starting points. In each case, the starting chord is $\{a,b,c\}$.

NRO	Major	Minor
R	$\{a,b,c+2\}$	$\{a-2,b,c\}$
P	$\{a,b-1,c\}$	$\{a,b+1,c\}$
L	$\{a-1,b,c\}$	$\{a,b,c+1\}$
N	$\{a,b+1,c+1\}$	$\{a-1,b-1,c\}$
N'	$\{a-2,b-2,c\}$	$\{a,b+2,c+2\}$
S	$\{a+1,b,c+1\}$	$\{a-1,b,c-1\}$

Procedural Neo-Riemannian Theory

The transformational relationship of NRT chord progressions is well-suited to work with visual, dialogue and interactive elements of a scene that are meant to evoke feelings. As such, it could be employed in generative systems to write expressive and associative music for films and games. Unfortunately, Cohn's (1998) mathematical description of NRT analytical techniques is frankly overly complicated and confusingly formalised in places. To prepare the theory for implementation in a generative system, we rationalise it here in terms of a set of *conditional rewrite rules*, R, which take a trichord and transform it to another. The conditional check is whether the starting chord is major or minor and each type of chord has six rewrite rules available to transform it, as per table 2. The table prescribes how to transform trichord $\{a,b,c\}$ into a new trichord by adding or subtracting a given number of semitones. As an example, rewrite rule R would take the C-major chord $\{C,E,G\}$ and transform it to $\{C,E,A\}$ (first row, first column of table 2), but would take the C-minor chord $\{C,E\flat,G\}$ and transform it to $\{B\flat,E\flat,G\}$ (first row, second column). Note that for a rewrite rules to be applied, a chord must be permuted into its prime form, i.e., with a being the tonic note, b being a major/minor third above a and c being a fifth above a.

Note that all the NROs rewrite a minor or a major chord to another minor or major chord. Hence, if we start with a major or minor chord, M, we can string together NROs to produce many more transformations of M than in table 2. For instance, starting again with the C-major chord, the sequence LPR would transform it as follows:

$$\{\mathrm{C,E,G}\}\xrightarrow[maj]{\mathrm{L}}\{\mathrm{B,E,G}\}\xrightarrow{\mathrm{I}}\{\mathrm{E,G,B}\}\xrightarrow[min]{\mathrm{P}}\{\mathrm{E,G\sharp,B}\}\xrightarrow[maj]{\mathrm{R}}\{\mathrm{E,G\sharp,C\sharp}\}$$

(Note the required step, I, to permute trichord inversion $\{B, E, G\}$ to the prime form $\{E, G, B\}$). This means that a generative system could transform chord $\{C, E, G\}$ into $\{E,G\sharp,C\sharp\}$ in three steps L, P then R but could also transform it directly in one step LPR.

This further means that a sequence of chords can be constructed using the guidelines of table 1 to follow the emotional/situational requirements of a given scene/cut scene in a game or film, or indeed to react live to player actions or other changes in a game. In addition, the length of the sequence could be used as an indication of the strength of emotional/situational change reflected by the chord transformation in the music, as per analytical NRT. This could be done in a stochastic way, perhaps driven by a Hidden Markov Model to introduce variety and surprise in the outputs. Finally, given that entire film soundtracks can be analysed in terms of sequences of NRO rewrite rules, it should be possible to drive the generation of new sequences via machine learning over a corpus of music from a particular composer, or from a particular film.

Related Work

The generation of chord progressions has been investigated for automatic generation of musical harmony, and Wiggins (1999) looks at the notion of intentionality in this respect, which is important for the generation of chord progressions to accompany film/game scenes. Bernardes et. al. (2016) implemented the D'accord harmony generation system which worked over a perceptually motivated tonal interval space. While not using NRT, Monteith et al. (2010) generated music to induce targeted emotions, using statistical techniques such as HMMs, and applied this in (Monteith et al. 2011) to produce affective music to accompany the audio of fairy tales being read.

Chew and Chuan (2011) proposed a style-specific accompaniment system that applies statistical learning to music theory frameworks including NRT. In this work, Neo-Riemannian Theory is used to represent the transitions between adjacent chords and NRT operators (NRO) are used on the Tonnetz, a conceptual lattice diagram representing the tonal space, to build decision trees to statistically learn melody-chord patterns. Given the styles this system mimics (e.g., Rock) and the roman numeral analysis (a type of music analysis where roman numerals are used to represent chords coordinating with scale degrees 1-7) used for the chord progressions, it is clear that this work aimed to create diatonic chord progressions using functional harmony, rather than triadic chromaticism. Similar limitations are seen in other related works such as Amram et al. (2020), where generative chord-based composition is implemented. Here the authors do not implement NRT to its full extent and consider atonality a disadvantage.

Conclusions and Future Work

We have provided a bridge from film and videogame music composition to computational creativity, via a description of the roles of music in films and games, and a procedural reading of an established music analysis technique, namely Neo-Riemannian Theory. The new formalism encapsulates four main points for generative music: (i) a set of rewrite rules for trichords (ii) the sequencing of rewrite rules to provide longer distance chord transforms (iii) a mapping of emotional and/or situational cues to sequences of rewrite rules, and (iv) the observation that chord transform distance is roughly proportional to the strength of the emotional change perceived in the music.

We are building a general-purpose music generation system specifically to aid with film and videogame compositions. At the heart of this will be procedural NRT, and we plan to draw further from music theory when applied to films, to supplement this approach. In particular, we will experiment with the automated invention, repetition and variation of short musical sequences called *leitmotifs*, (Whittall 2001), which can represent characters, emotions, locations and other elements in a film or game. Our aim is to use AI techniques such as HMMs, deep learning, constraint solving and planning to harness NRT and leitmotifs into a system of real utility for composers.

The system will take as input tags that describe emotions, allowing the output to directly follow the scene's emotional arc or, in some less canonical soundtrack examples, provide a contrast to the emotions seen in the media (as suggested by an anonymous reviewer). Producing film and videogame music that deconstructs the viewer's expectations by representing a different emotion than the one seen on screen can create a powerful viewing experience. An example of this technique can be seen in the psychological horror movie Us, where the composer uses a mixture of well-known, happy, upbeat songs to provide a terrifying contrast from the violence that is happening on-screen. The use of unexpected songs which provide a contrasting mood from the visuals can heighten the feelings of fear as they do not offer any hint on what is going to happen, as most canonical soundtracks do. Other times, contrasting moods between music and visuals can be used to provide a comical effect, such as using a feel-good song while a character is going through hardships. We plan to take into account such less-canonical scoring techniques by providing a system that can create soundtracks that respond or contrast the visuals. As aforementioned, this could be achieved by manually annotating the film with mood tags, so that the composer or director can choose how the visuals should be represented, or not represented, by the music in order to create a variety of interesting viewing experiences for the audience.

Future projects include a collaboration with Younès Rabii, developer of the videogame Tea Garden (Pyrofoux 2020). We plan on applying an NRT-based generative system to produce music for games where characters are created dynamically, such as tabletop role-playing games or videogames that feature a component of live automated videogame design (Rabii and Smith Nicholls 2022). Neo-Riemannian Theory would provide an effective framework to produce music that quickly responds to changes in the characters and videogame design, given the possible combinations of Neo-Riemannian operators and the events and emotions they can represent.

We also plan to use our NRT generative system alongside @artbhot (Smith and Colton 2022), a Twitter bot that generates images from user-given text prompts, to create multimedia outputs such as stories with background music. We plan on exploring using music as a background for a story created from a user-given prompt, or creating music from a user-given prompt and using visual imagery to accompany it. As NRT can be used to write music in various genres, it should produce music suitable to represent the story line or the user-given prompt.

Author Contributions

Theoretical work on the rationalisation of NRT for generative purposes was undertaken by the first author, Sara Cardinale, with help from the second author Simon Colton. All authors participated in the writing of this manuscript, with the first author taking the lead role in this.

Acknowledgements

This work has been funded by UKRI and EPSRC as part of the "UKRI Centre for Doctoral Training in Artificial Intelligence and Music", under grant *EP/S022694/1*. We would like to thank the anonymous reviewers for providing helpful and friendly feedback, which has improved this paper.

References

Amram, M.; Fisher, E.; Gul, S.; and Vishne, U. 2020. A transformational modified Markov process for chord-based algorithmic composition. *Mathematical and Computational Applications* 25(3).

Bernardes, G.; Cocharro, D.; Guedes, C.; and Davies, M. 2016. Harmony generation driven by a perceptually motivated tonal interval space. *Computers in Entertainment* 14(2).

Chuan, C.-H., and Chew, E. 2011. Generating and evaluating musical harmonizations that emulate style. *Computer Music Journal* 35:64–82.

Cohn, R. 1998. Introduction to Neo-Riemannian Theory: a survey and a historical perspective. *Journal of Music Theory* 42(2).

Huron, D. 2006. *Sweet Anticipation: Music and the Psychology of Expectation*, volume 1. MIT Press.

Hutchings, P. E., and McCormack, J. 2020. Adaptive music composition for games. *IEEE Transactions on Games* 12(3):270–280.

Lehman, F. 2013. Transformational Analysis and the Representation of Genius in Film Music. *Music Theory Spectrum* 35(1):1–22.

Lehman, F. 2014. Film music and NRT. *Oxford Handbook*.

Lehman, F. 2018. *Hollywood Harmony: Musical Wonder and the Sound of Cinema. Oxford University Press*.

Lewin, D. 1982. A formal theory of generalized tonal functions. *Journal of Music Theory* 26(1):23–60.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2014. Computational game creativity. In *Proceedings of the International Conference on Computional Creativity*.

Monteith, K.; Francisco, V.; Martinez, T.; Gervás, P.; and Ventura, D. 2011. Automatic generation of emotionally-targeted soundtracks. In *Proceedings of the International Conference on Computional Creativity*.

Monteith, K.; Martinez, T.; and Ventura, D. 2010. Automatic generation of music for inducing emotive response. In *Proceedings of the International Conference on Computational Creativity*.

Plut, C., and Pasquier, P. 2020. Generative music in video games: State of the art, challenges, and prospects. *Entertainment Computing* 33:100337.

Plut, C. 2017. *The Audience of the Singular (MFA Thesis)*. Simon Fraser University, Vancouver, Canada.

Rabii Younès (Pyrofoux) 2020. Tea Garden. Available at: https://pyrofoux.itch.io/tea-garden.

Rabii Younès (Pyrofoux) and Smith Nicholls Florence. 2022. Choose Your Own Misadventure: AI-Powered Futures for Game Design. Talk given at Game Developers Conference 2022. Available at: https://www.youtube.com/watch?v=qYxRToBWpVo &t=2466sab_channel=Knives%26Paintbrushes.

Rehding, A.; Floud, R.; A,; Johnson, P.; Kallberg, J.; Newcomb, A.; and Solie, R. 2003. *Hugo Riemann and the Birth of Modern Musical Thought*. Cambridge University Press.

Rod, M. 2007. *Music In Video Games, in J. Sexton (ed.) Music, Sound and Multimedia: From the Live to the Virtual*. Edinburgh University Press.

Savery, R., and Weinberg, G. 2020. Shimon the robot film composer and deepscore: An LSTM for generation of film scores based on visual analysis. *arXiv* 2011.07953.

Smith A., and Colton S. 2022. The @artbhot Text-To-Image Twitter Bot. In *Proceedings of the International Conference on Computational Creativity*.

Whalen, Z. 2004. Play along – an approach to videogame music. *The International Journal of Computer Game Research* 4.

Whittall, A. 2001. *Leitmotif*. Oxford University Press.

Wiggins, G. A. 1999. Automated generation of musical harmony: What's missing. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

Connecting Audio and Graphic Score Using Self-supervised Representation Learning - A Case Study with György Ligeti's Artikulation

Berker Banar and Simon Colton
School of Electronic Engineering and Computer Science
Queen Mary University of London, UK
{b.banar, s.colton}@qmul.ac.uk

Abstract

Music is a phenomenon that can be represented in various data modalities, such as MIDI, musical score, graphic score and audio. Connecting these modalities in an informative and intelligent way is important, especially for multi-modal music generation systems. In this study, we present a novel self-supervised representation learning approach that can be applied to finding a mapping between audio and graphic scores in a generative context. Our approach consists of two variational autoencoder-based generators and a contrastive learning mechanism. We demonstrate this technique using György Ligeti's Artikulation, which is an electronic music composition with a graphic score. In initial experiments, given manually designed graphic score excerpts in the style of Artikulation, we generated good quality audio correspondents with our model. We further suggest some ways of improving our approach and discuss some future directions for our work.

Introduction

Music can be represented in audio and symbolic (e.g. musical score, MIDI, or graphic score) domains. Commonly in generative music studies, just one of these data modalities is targeted and the generative system is specifically designed for the selected domain (Oord et al. 2016) (Payne 2019). While there are studies that focus on connecting some of these modalities (Wang and Yang 2019), the full potential of multi-modal representations has not been fully explored in generative contexts yet. This is especially true for a wide range of sonic and timbral options and various music representations. Connecting these different music representations is beneficial in an end-to-end multi-modal music generation pipeline, where the generation starts in the symbolic music domain, and then symbolic material is converted into audio via a mapping between symbolic and audio representations. In this multi-modal setting, we benefit from the advantages of both worlds, where the symbolic representation enables us to control the generation process in terms of some high-level musical attributes such as tonality, harmony and rhythmic complexity, and provides us with a confined format; while the audio representation allows us to introduce expressive, textural and complex elements in a sonic domain, where we appreciate music as people.

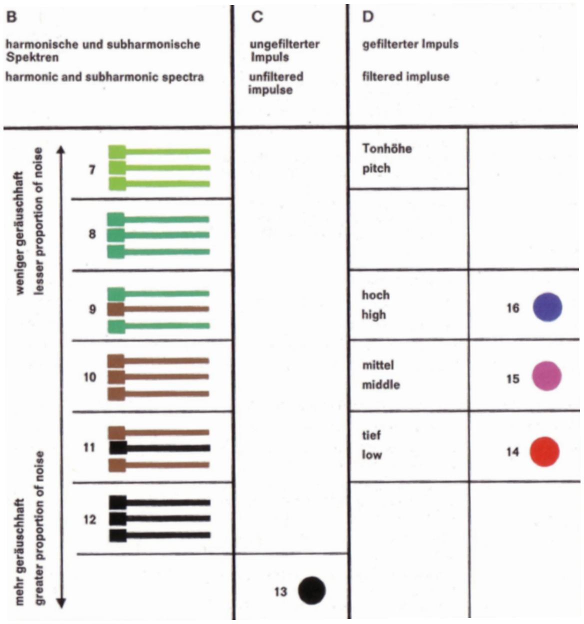


Figure 1: Legend for sonic objects in the graphic score of György Ligeti's Artikulation.

It has become clear recently that self-supervised representation learning can be highly effective, as highlighted by the success of the CLIP model (Radford et al. 2021) for mapping both images and text into the same latent space. Such contrastive learning can then be used in generative methods, for instance with CLIP being used to guide GAN image generation, such as with BigGAN (Brock, Donahue, and Simonyan 2018) or VQGAN (Esser, Rombach, and Ommer 2021). Inspired by these successes, we believe self-supervised representation learning approaches for connecting symbolic music and audio domains could enhance the creative potential of generative music models.

In the matter of symbolic music representations, traditional musical scores might be limited in terms of expressing the actual music itself, specifically in scenarios such as electroacoustic and acousmatic music. In contemporary classical music (Spencer 2015), graphic scores act as alternative music notations, and allow more expressive performance details to be represented, particularly for subtle and continuous

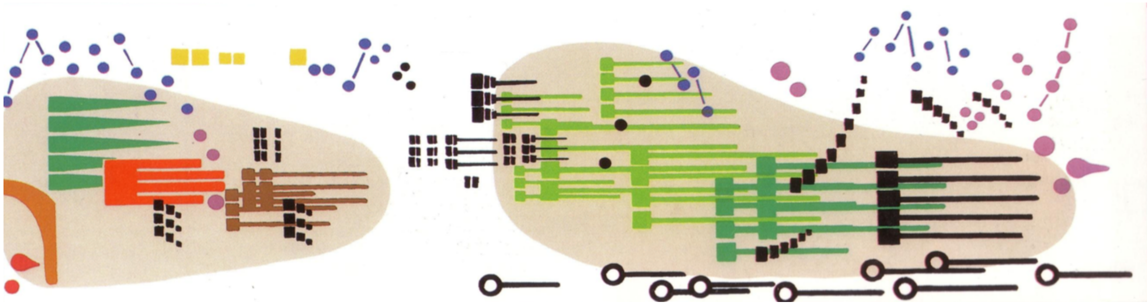


Figure 2: Graphic score fragment from Ligeti's Artikulation.

changes. Such scores engage performers to follow abstract visual mappings, which can be attractive to manipulate for inexperienced practitioners, e.g., for those without formal training on traditional Western scores. Graphic scores are not universal, however, and their organisation depends a lot on the unique mappings given in a legend, as in Figure 1; deciphering graphic scores is a challenging task. One well-known graphic score is for Artikulation by György Ligeti, designed by Rainer Wehinger, who first listened to the piece and then constructed coherent abstractions to illustrate the musical entities presented. An example of this score is depicted in Figure 2. In the organisation of this graphic score, the horizontal axis represents time, the vertical axis represents pitch and coloured shapes represent unique sonic entities that are used in the piece as themes and musical ideas.

In this study, we present a self-supervised representation learning framework to connect audio and graphic score domains, and demonstrate a creative composition use case that allows practitioners to compose in the style of Artikulation utilising its visual and sonic universe. Without such an approach, this task might not be possible, as it is challenging to separate and re-synthesize the complex textures in Artikulation by listening to the piece and looking at the graphic score and its abstract legend. Practically, in our use case, our system allows us to generate new audio segments, which are conditioned on manually created graphic score excerpts that are not part of the original graphic score, but in its graphical style. Demonstrating this feature, we exhibit some manually created graphic score fragments and their synthesised correspondents within the aesthetics of the piece. To conclude, we address potential ways of improving this system and some future directions for this study.

Data Processing

The original graphic score of Artikulation is presented in fragments of 5 to 10 seconds duration. First, we cropped these fragments and manually processed them to get rid of the time axis lines and canvas contours, then merged these processed fragments into a single long image file constituting the whole graphic score for the piece. Then, we extracted graphic score excerpts using 2 seconds of windows, where the stride amount is 1 second. As the piece is 227 seconds long, this excerpt extraction process gave us 226 windows in total. Then, we further processed these extracted images to restrict their palettes to 10 discrete colours to make the learning procedure easier. One caveat is that this procedure gets rid of the grey shaded regions in the original score, which represent the effect of reverb. In our future work, we will further experiment with graphic score excerpts that have such reverb regions.

We recorded the audio file of Artikulation while streaming the piece online from YouTube at 44.1kHz sampling rate and applied a similar data processing where 2 seconds of audio fragments were extracted, again with the stride amount of 1 second. These audio fragments were paired with their corresponding graphic score excerpts. Then, we used constant-q transform (CQT) (Schörkhuber and Klapuri 2010), which is a wavelet-based time-frequency transform, to generate spectrograms for each audio file, to be used in the learning process.

Model Architecture

Our architecture consists of three main sub-parts, which are an audio pipeline, a graphic score pipeline, and a contrastive learning block for self-supervised representation learning, as illustrated in Figure 3. Both the audio and graphic score pipelines utilise a variational autoencoder (VAE) architecture (Kingma and Welling 2013) and our contrastive learning mechanism is based on the cosine similarity between audio/graphic score latent representations using a duplet loss.

In the audio pipeline, we have an encoder-decoder architecture, which is taken from (Tatar, Bisig, and Pasquier 2021) and the audio data is presented to the network in CQT spectrogram format (Schörkhuber and Klapuri 2010). The encoder part has two consecutive 4096-dimensional dense layers that are followed by two parallel 4096-dimensional dense layers embedding in two 512-dimensional spaces, which are for the mean and the variance of variational sampling to a 512-dimensional space. The decoder part has three dense layers with 4096 dimensions. During the training procedure, we use the Adam optimiser (Kingma and Ba 2014) where the learning rate is 0.0001, β_1 is 0.9 and β_2 is 0.999. As per the typical configuration of VAEs, the loss function of this encoder-decoder architecture has two parts, namely the reconstruction loss and regularisation loss, and a mean squared error loss function is used for the reconstruction part, where KL-divergence (Kullback and Leibler 1951) is used for the regularisation. In this pipeline, our decoder generates CQT spectrograms, which are then converted into audio files using fast Grifin-Lim phase reconstruction as in (Tatar, Bisig, and Pasquier 2021).

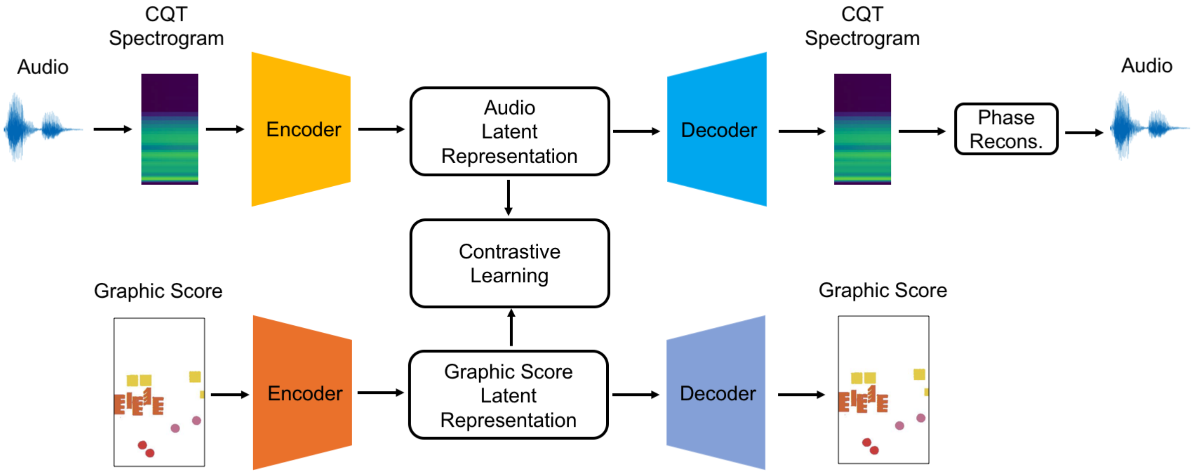


Figure 3: Architecture schematic for the two VAEs and contrastive learning block.

The graphic score pipeline also uses an encoder-decoder architecture, which directly takes and generates RGB images on both sides. The encoder here first flattens the RGB image, then passes it through a 2048-dimensional dense layer, which is followed by two parallel 2048-dimensional dense layers. The embedding space has 512 dimensions similar to the audio pipeline. The decoder part consists of two 2048-dimensional dense layers, which are followed by a deflattening procedure, which converts single stream decoder outputs into three channel RGB images. An adam optimiser is used as in the audio pipeline with the same parameters. Similar to the case above, we use mean squared error and KL-divergence for reconstruction and regularisation losses, respectively.

The contrastive learning block aims to make the corresponding embeddings of the graphic score and audio pairs as close to each other as possible, using cosine similarity between their mean and variance latent vectors in variational autoencoders. The training procedure utilises a multi-task optimisation process, where we train the VAE architectures for reconstruction and the contrastive learning block for self-supervised representation learning using a unified loss, simultaneously. Since we have two main objectives, which are the reconstruction quality of VAEs and creating a structured embedding space, we weight our VAE and contrastive losses. Based on our initial experiments, which suggest that audio reconstruction might be a more challenging task in this setting and require more attention, VAE losses for the graphic scores and audio are weighted as 10% and 90% with respect to each other. We also downscale the cosine similarity loss between 512-dimensional latent vectors by a factor of 50, in order to make it aligned with the VAE losses and have 0,0x decimal numbers for each of our losses at the beginning of our training procedure. We trained our complete model for 200 epochs with a batch size of 32.

We use this architecture in a multi-modal generative setting, where a user-designed graphic score is encoded into the embedding space and its latent vector is decoded using the audio decoder. Graphic score and audio embeddings share the same latent space due to the contrastive learning strategy, thus, the latent vector of a given graphic score can be interpreted keeping the semantic connections between two data modalities. This shared embedding space approach has been successfully demonstrated in the CLIP model (Radford et al. 2021), which uses text and image data, but CLIP requires a separate generator to create artefacts. In our approach, we combine the self-supervised representation learning and generation tasks in the same model and training procedure, and utilise this technique with graphic scores and audio, which allow us to create a generative universe in the style of a piece or a composer.

Experiments

To demonstrate the reconstruction capability of our model, we use four audio and four graphic score excerpts that are all from Artikulation, originally. We reconstruct these excerpts using our audio and graphic score pipelines that are trained separately without the contrastive learning block. We also reconstruct these original excerpts using our trained complete architecture. All of the reconstructed graphic scores including the originals are exhibited online (Figure 5, 6 and 7)[1] and all the reconstructed and original audio files (Original 1-4) are presented on a SoundCloud page[2]. For the separately trained pipelines, as demonstrated with the figures and audio files, reconstruction quality is high. For our trained complete architecture, although the reconstruction quality slightly decreases compared to the separately trained pipelines, which is expected due to introducing the contrastive learning block, reconstructed graphic scores and audio files exhibit intelligible graphical objects and good quality sonic entities, respectively.

In order to test our trained architecture in a multi-modal generation scenario, we manually designed four different graphic score fragments in the style of Artikulation, which are not exactly the same as any of the original graphic score fragments. This approach demonstrates the creative potential of the system, where creators can compose their own

[1] https://bit.ly/37A1CgV

[2] https://soundcloud.com/user-330551093/sets/audio-sym-ssrl

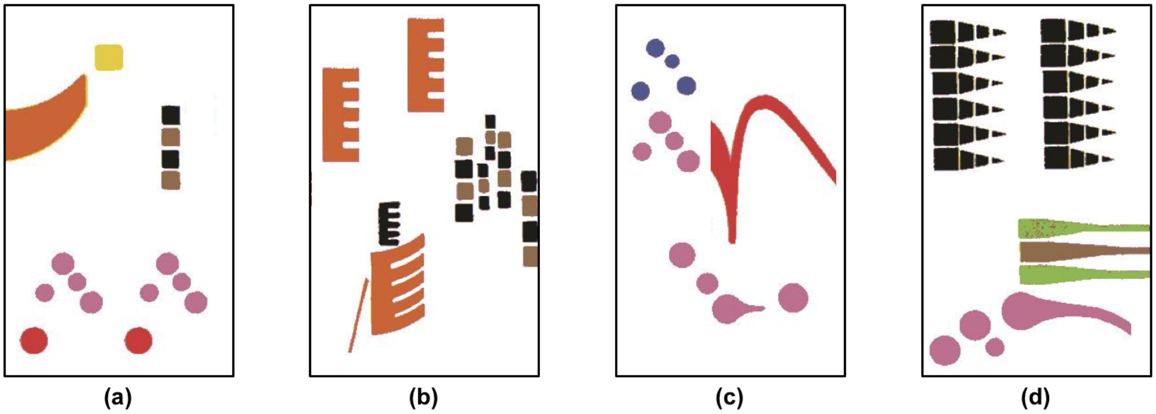


Figure 4: Four manually designed graphic score excerpts.

musical pieces by designing graphical scores in the universe of Artikulation. Our expectation is that combined models are able to generate an audio excerpt that sonically reflects the material presented in the given graphic score fragment in alignment with the characteristics of Artikulation. Our designed graphic scores are presented in Figure 4 and we display their reconstructed versions generated using our complete model via the same link[1] (Figure 8). Even though the reconstructed versions are lower in quality compared to the originals, they are successful in terms of representing the graphical content, shapes and colours. We exhibit the multi-modally generated audio files of four manually designed graphic score excerpts on the same SoundCloud page[2]. When we analyse these audio files, even though they are not considered to be good quality regarding clear sonic textures compared to Artikulation, the generated audio files still exhibit the textures of the piece and are reflective in terms of the visual composition.

When we evaluate these audio files in more detail, in the beginning of generated audio for graphic score (a), we have a sonic element with rising pitch similar to the curved orange shape on its graphic score. Generated audio for excerpt (b) demonstrates a similar rising pitch object, but lower in pitch compared to excerpt (a), which can be associated with the comb-shaped curvy figure on the lower side of the vertical axis, which corresponds to musical pitch. In the audio excerpt for graphic score (d), we have a unique and strong sonic statement, which might be reflected in the horizontal cone-like black figures. A similar sonic entity is repeated on the second half of this audio excerpt, but differently, which might correspond to the second set of black figures. The difference can be due to having horizontal green and brown shapes happening at the same time. In our future work, we plan to quantitatively analyse generated audio files using audio similarity metrics, to better evaluate the reflectiveness of their given graphic scores and the style of Artikulation in general.

Conclusions

In this study, we present a novel framework that connects audio and graphic score domains using self-supervised representation learning, which can be extended to other music data modalities. We demonstrate its use case in a scenario where we utilise Ligeti's Artikulation represented in both a graphic score and audio forms, and also exhibit the results of our initial experiments with the system, which generates music in audio format in the style of Artikulation based on unseen but stylistically similar graphic score excerpts presenting a creative use case of this generative system in the context of human-machine co-creation. Even though the results are not perfect, we believe that this approach has valuable potential, especially to be utilised in multi-modal music generation systems. Also, due to the artistic form of this graphical music representation, we think that sonifying visuals in a defined sonic and visual space is valuable from a computational creativity perspective, as it might allow to further pieces with rich textures referencing a variety of visual abstractions and reflecting complex styles of composers.

In our future work, we plan to improve the quality of the generated material as well as the generalisation capability of the model by further experimenting with the architecture and applying data augmentation both in visual and audio domains. Also, besides our own subjective evaluation, we will introduce numerical metrics which can evaluate the closeness of generated audio material to given graphic scores in the context of Artikulation. To improve the match between a given graphic score excerpt and its corresponding generated audio, we plan to experiment with introducing various inductive biases to the model, which might ease the learning process and allow the model to learn a mapping between the graphic score and audio more effectively. Besides Artikulation, we will experiment with other contemporary classical music pieces with graphic scores using our approach. Additionally, we are interested in using this technique in other combinations of data modalities as well, such as audio-MIDI. Moreover, we would like to build an online tool based on this system, which can generate music using graphic score excerpts specifically created by the users. Furthermore, we plan to utilise this system in a scenario where an audio excerpt in the style of Artikulation is provided and the model is expected to generate its corresponding graphic score (i.e., the reverse direction to the inference workflow discussed here), which enhances the creative potential of this approach.

Author Contributions

The majority of the technical and experimental work for this paper was undertaken by the first author, Berker Banar. The second author, Simon Colton, contributed some image processing code and advice on the direction of this work. The paper was written by Berker Banar with some contributions and editing by Simon Colton.

Acknowledgments

Berker Banar is a research student at the UKRI Centre for Doctoral Training in Artificial Intelligence and Music, supported jointly by UK Research and Innovation [grant number EP/S022694/1] and Queen Mary University of London. We wish to thank the anonymous reviewers for their insightful comments.

References

Brock, A.; Donahue, J.; and Simonyan, K. 2018. Large scale gan training for high fidelity natural image synthesis. *arXiv:1809.11096*.

Esser, P.; Rombach, R.; and Ommer, B. 2021. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12873–12883.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv:1412.6980*.

Kingma, D. P., and Welling, M. 2013. Auto-encoding variational bayes. *arXiv:1312.6114*.

Kullback, S., and Leibler, R. A. 1951. On information and sufficiency. *The Annals of Mathematical Statistics* 22(1):79–86.

Oord, A. v. d.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.; and Kavukcuoglu, K. 2016. Wavenet: A generative model for raw audio. *arXiv:1609.03499*.

Payne, C. 2019. Musenet. https://openai.com/blog/musenet/. Accessed: 2022-04-20.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, 8748–8763.

Schörkhuber, C., and Klapuri, A. 2010. Constant-q transform toolbox for music processing. In *Proceedings of the 7th sound and music computing conference, Barcelona, Spain*, 3–64.

Spencer, M. 2015. Art and music collide in these 20 stunning graphic scores. https://www.classicfm.com/discover-music/latest/graphic-scores-art-music-pictures/. Accessed: 2022-04-20.

Tatar, K.; Bisig, D.; and Pasquier, P. 2021. Latent timbre synthesis: Audio-based variational auto-encoders for music composition and sound design applications. *Neural Computing and Applications* 33(1):67–84.

Wang, B., and Yang, y.-h. 2019. PerformanceNet: Score-to-audio music generation with multi-band convolutional residual network. *Proceedings of the AAAI Conference on Artificial Intelligence* 33:1174–1181.

Co-creation and ownership for AI radio

Skylar Gordon[1], Robert Mahari[12], Manaswi Mishra[1], Ziv Epstein[1]
{sfgordon, rmahari, manaswim, zive}@mit.edu
[1] MIT Media Lab [2] Harvard Law School

Abstract

Recent breakthroughs in AI-generated music open the door for new forms for co-creation and co-creativity. We present Artificial.fm, a proof-of-concept casual creator that blends AI-music generation, subjective ratings, and personalized recommendation for the creation and curation of AI-generated music. Listeners can rate emergent songs to steer the evolution of future music. They can also personalize their preferences to better navigate the possibility space. As a "slow creator" with many human stakeholders, Artificial.fm is an example of how casual creators can leverage human curation at scale to collectively navigate a possibility space. It also provides a case study to reflect on how ownership should be considered in these contexts. We report on the design and development of Artificial.fm, and provide a legal analysis on the ownership of artifacts generated on the platform.

Introduction

"As notions about the nature and function of music become embedded in the structure of software-based musical systems and compositions, interactions with these systems tend to reveal characteristics of the community of thought and culture that produced them."
–George Lewis (Lewis, 2000)

Recent breakthroughs in deep learning have introduced the opportunity for generating high-fidelity songs in the raw audio domain. Some believe that this new potential portends the end of musical creativity, while others think it represents yet another tool to augment musical production. Both sides have merit, but of particular importance is the fact that these end-to-end music generation systems can synthesize music without any understanding in music composition or technique. This opens up the possibility of embedding them in computational creativity systems, which allows users to explore a large possibility space of music without formal musical training(Compton and Mateas, 2015) and engage in *mixed-initiative co-creativity* Yannakakis, Liapis, and Alexopoulos (2014).

To interrogate these questions, we introduce a proof-of-concept casual creator system, Artificial.fm, that allows listeners to help curate and steer the evolution of music generated with OpenAI's Jukebox model (Dhariwal et al., 2020). In addition to listening to this new kind of music, listeners can also provide feedback on the generated songs, thus helping the AI learn to generate better music in the future. The system also uses these ratings to provide personalized music recommendations, which helps the music evolve to fit the preferences of the listener.

These components form an interconnected sociotechnical system for music generation and curation, with many distinct human stakeholders. This distributed model of production complicates the definition of *the user* of the system, since many different users are involved in different ways. It also raises important questions about who owns the artifacts generated by the system.

Our system falls in the lineage of "slow creators" defined by a "problematic gulf of execution" (Compton, 2019). This collection of creators involves most audio-based generators, since evaluating songs requires the user to actually listen to the outputs, instead of quickly discerning its quality, as with visuals. However, a distinct yet understudied aspect of Artificial.fm is the fact that generation itself is high-latency and therefore impossible to do on the fly: Jukebox takes about 20 hours to generate 20 seconds of audio. Thus Artificial.fm explores design patterns for a growing set of systems where intensive underlying computation means real-time interaction with the underlying generator is fundamentally infeasible.

In this paper, we present the case study of Artificial.fm to highlight how slow creation can translate to the evaluation and curation of AI-generated music. We then use legal precedent to trace the multiple stakeholders involved in this process and unpack the each actor's stake in ownership.

Related Work

Algorithmic Music has a rich history amongst composers starting in the pre-computing era from the process works of George Brecht's Drip Music (1962), Stockhausen's Setz die Segel zur Sinn (Maconie, 1970) and Xenakis' Formalised Music (Xenakis, 1992) to the formation of the US League of Automatic Music Composers (1978).

Algorithms have been used to generate music both in the symbolic domain (Hiller Jr and Isaacson, 1957; Moorer, 1972; Hadjeres, Pachet, and Nielsen, 2017; Huang et al., 2018) and in the waveform domain through digital vocoders (Bonada and Serra, 2007; Blaauw and Bonada, 2017) and synthesizers (Mehri et al., 2016; Engel et al., 2017).

Compton (2019) identifies Musical computational creativity systems as inherently "slow creators" where the user evaluation in the *grokloop* is implicitly slow. Though computationally slow in generating these more complex musical generative spaces, this newer generation of 'slow systems' are capable of producing more aesthetically pleasing and uniquely shaped outputs that often feel more rewarding and personal to the user. More recent casual creators like - Magenta's Tone Transfer, Piano Genie and applications produced at the BitRate ML and Music hackathon 2020, take advantage of modern AI models to produce higher fidelity musical outputs.

These generators are characterized by large possibility spaces, which can be difficult for individuals to explore. A promising approach to rapidly search through a large possibility space to find the "gem in the rough" is to crowdsource its exploration. A diverse set of casual creators leverage collaborative media to produce intriguing artifacts. The Reddit R/Place experiment had users collaboratively paint a pixel canvas (Rappaz et al., 2018). Drave's Electric Sheep used user feedback and evolutionary algorithms to generate new "sheep" - fractal animations adapted to crowd preferences (Draves, 2005). PicBreeder also uses evolutionary algorithms and allows users to collaboratively evolve images Secretan et al. (2008, 2011). Feed the Ganimals allowed users to explore and curate AI-generated hybrid animals, and found that social cues led to the formation of diverse local trends (Epstein et al., 2020a, 2021).

System Overview

Artificial.fm uses OpenAI's Jukebox (Dhariwal et al., 2020), a deep neural network trained on 1.2 million songs, for music generation. Jukebox has the ability to take as input a "prime" of existing music which it then improvises on top of. We solicit primes from local musicians as part of a collaboration to support artists affected by the pandemic. Jukebox also takes in a specified artist and genre as inputs which condition the style of the generated song outputs.

The outputs of the song generation process are streamed via the platform, where listeners provide subjective feedback on the AI-generated music, in the form of ratings. The questions related to how happy, danceable, artificial, instrumental, upbeat and song was, and how clear the lyrics, and if they liked it, on a 5-point Likert scale (see Supplementary Information Section 1.2 for more information).

The crowdsourced feedback is then used to adapt the generation process with an algorithm that balances exploring new permutations of parameters with exploiting existing parameters that are popular with users. This is achieved using a variation of Thompson sampling, which is regret-minimizing in such contexts (Chapelle and Li, 2011). To do so, we use the Spotify API and Essentia (Bogdanov and others, 2013) to generate a rich set of covariates for the artist of the prime, as well as candidate artists and genres (see Supplementary Information Section 1.3 for more information on how these covariates are generated).

As new primes are solicited from local musicians, the following algorithm finds parameters (e.g. an artist and genre prompt) to pair with that prime to balance exploration and exploitation: First, we fit a model $\hat{f}$ predicting ratings of the existing songs (e.g. How much do you like this song?) based on the Spotify covariates of that song's prime artist, artist prompt, and genre prompt (27 features total). Then, we sample M artist, genre pairs from the joint distribution of these prompts in the input space. Then, we predict the rating of that artist, genre pair for the given prompt $\hat{f}_{prime}(a_\ell, g_\ell)$. We then take the top γ artist, genre pairs and randomly sample one uniformly (here γ controls exploration vs exploitation, $\gamma = 1$ is maximal exploitation, $\gamma = M$ is maximal exploration). See Supplementary Information Section 1.4 for more details about this algorithm. [1]

Artificial.fm also provides personalized song recommendations to users. Through a preference elicitation interface (see Figure S3), users can explicitly specify the kind of songs they would like to hear. Based on their stated preferences, a personalized recommendation algorithm serves songs to them consistent with these preferences (see Supplementary Information Section 1.5 for more details about this recommender system).

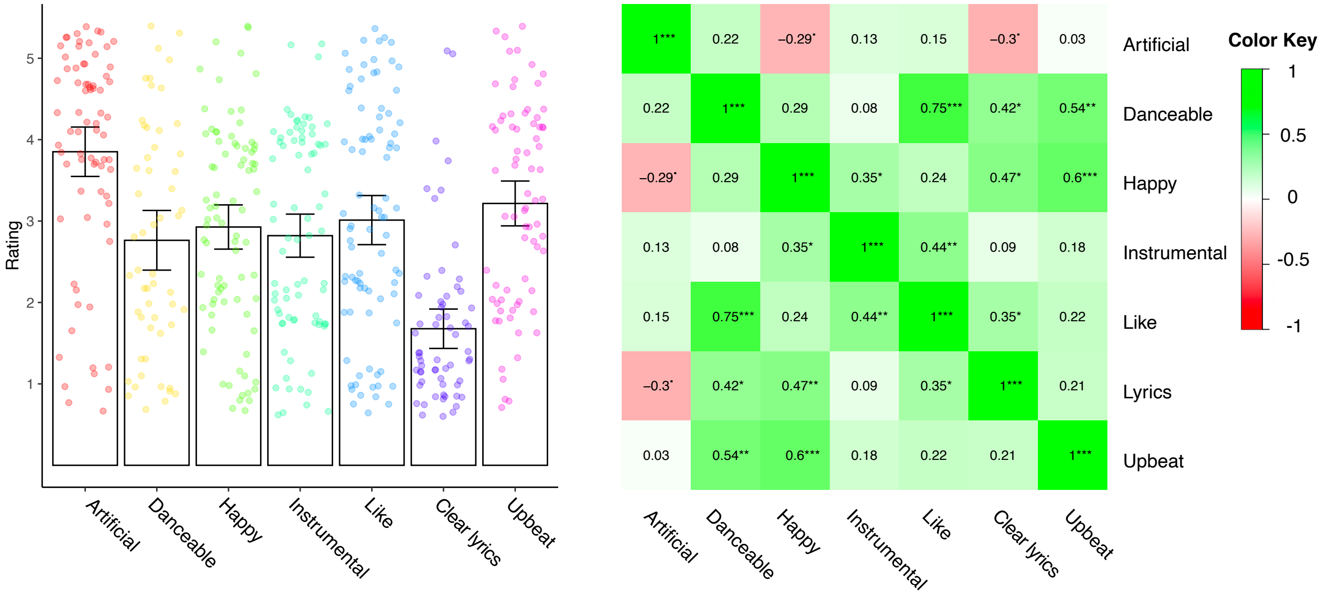
Data and Results

As of July 21, 2021, we accumulated 522 ratings of 71 songs by 40 people. The songs were generated with genre prompts from folk, house, pop, americana, rock, classical, electronic, and funk, and artist prompts from The Weeknd, Aerosmith, The Doors, Justin Bieber, Elton John, Dolly Parton, Otis Redding, and Lady Gaga. The primes were sourced from several local artists we reached out to. The 40 people found the platform through word of mouth.

The distribution of ratings by question is shown on the left of Figure 1. Relative to the other questions, listeners found the songs highly artificial (one-sided t test, $p < 0.001$), and lacking in clear lyrics ($p < 0.001$). This suggests that the music of Artificial.fm may not fall into the "normal distribution" of what you find on the radio, but instead represents a polyphonous new kind of music onto itself. That being said, we did find meaningful variation in all seven questions ratings, which suggests there is quantifiable diversity in the possibility space to explore and optimize.

The pairwise correlations between these seven questions are shown on the right of Figure 1. We observe that perceptions of liking a song is associated with ratings of a song being danceable ($R = 0.75$, $p < 0.001$), instrumental ($R = 0.44$, $p = 0.004$), and having clear lyrics ($R = 0.35$, $p = 0.037$). We also find that ratings of the artificiality of a song are marginally negatively associated with having clear lyrics ($R = -0.30$, $p = 0.088$), and how happy the song is perceived to be ($R = -0.29$, $p = 0.078$).

[1] This algorithm assumes there is already a large number of both songs and ratings and therefore requires solving the "cold-start problem." Since the scope of this short paper is introducing the concept of AI radio via Artificial.fm with preliminary user testing and ethical considerations, this algorithm should be considered as a sketch for how Artificial.fm would work at scale. As such, we leave formal evaluation of such an approach to music generation to future work.



	Artificial	Danceable	Happy	Instrumental	Like	Clear lyrics	Upbeat
Artificial	1***	0.22	–0.29˙	0.13	0.15	–0.3˙	0.03
Danceable	0.22	1***	0.29	0.08	0.75***	0.42*	0.54**
Happy	–0.29˙	0.29	1***	0.35*	0.24	0.47*	0.6***
Instrumental	0.13	0.08	0.35*	1***	0.44**	0.09	0.18
Like	0.15	0.75***	0.24	0.44**	1***	0.35*	0.22
Lyrics	–0.3˙	0.42*	0.47**	0.09	0.35*	1***	0.21
Upbeat	0.03	0.54**	0.6***	0.18	0.22	0.21	1***

Figure 1: Left: Song ratings by question. Right: Pairwise correlation matrix between question ratings. $^{\cdot}$ refers to $p \leq 0.1$, * refers to $p \leq .05$, ** refers to $p \leq .01$, *** refers to $p \leq .001$.

Ownership of AI-Generated Music

The owner of a casual creators' output should be the entity responsible for creation. Compton and Mateas (2015) defines casual creation as the "the creation [of] new artifacts that bring feelings of pride, ownership, and creativity to the users that make them." This definition centers the users of a casual creator as the owners of its output. Artificial.fm challenges this idea of ownership and highlights open questions related to ownership of AI-generated works. [2]

At least five actors could claim some level of ownership over the works created by Artificial.fm: (1) the artist who submitted the prime on which a piece of music is based, (2) the many artists whose music was used to train Jukebox, (3) the system architects who developed Artificial.fm, (4) the listeners whose ratings are used to steer music production, and (5) the artificial intelligence itself. This section begins to explore the question of ownership for casual creators by analyzing the legal basis on which these actors may claim ownership and concludes by suggesting ownership models better suited to the distributed nature of systems like Artificial.fm.

The Prime and Training Artists 🧑‍🎤👩‍🎤

Both the prime and training data artists could claim ownership over a given piece of music created by Artificial.fm by arguing that Artificial.fm infringes on their copyright. To this end, they would need to show that the generated music is "substantially similar" to their work (Williams v. Gaye, 2018) and that the music was not independently created (Feist v. Rural Telephone Service Co., 1991).

The prime artists explicitly provide direct access to their works but the training data artists do not. Moreover, it is unclear whether Artificial.fm has "access" to the underlying training data because the music in the training data has been transformed into the Jukebox algorithm which does not contain copies of the works it has been trained on. Even if Artificial.fm has access to the artists' work, an infringement claim would require showing that a song created by Artificial.fm is substantially similar to a given artist's work (Williams v. Gaye, 2018). Substantial similarity is assessed using a two part test: first, an objective test where a music expert analytically compares the elements of two works for substantial similarity and second, a subjective test where an "ordinary reasonable person" assesses if the two works feel substantially similar (Swirsky v. Carey, 2004). Different experts and "ordinary people" may disagree about substantial similarity making these tests inherently vague. In the Artificial.fm case, it is likely that some generated music is similar to some works owned by prime artists, but it is unlikely for generated music to be substantially similar to songs in the training data.

The System Architects and Listeners 👩‍🔬🙋‍♂️

The system architects and listeners play their own role in creating the output of Artificial.fm and could claim ownership over the generated content. To focus on their contribution, imagine that Artificial.fm was trained exclusively on works in the public domain.

On one hand, the system architects might be akin to photographers who compose photographs by documenting objects from the real world. The U.S. Supreme Court clari-

[2] We use the term "ownership" broadly to encompass all the rights commonly associated with authorship. Where relevant, this section will base its analysis on U.S. and California law.

fied in 1884 that photography is to be treated as an art under copyright law, and that the photographer is to be treated as the "mastermind" whose creativity gives rise to a copyrightable work (Burrow-Giles Lithographic Co. v. Sarony, 1884). The system architects can similarly be characterized as the masterminds, who use their ingenuity to take advantage of a technology to produce works of art. On the other hand, although the listeners are using a tool built by the system architects, it is the listeners' preferences, not the architects', that guide what Artificial.fm produces. In this sense, the listeners are akin to photographers and the system architects are similar to camera makers, who have no claim to the photographs made with the technology they built.

Along these lines, the AI Artist Mario Klingemann often refers to himself as a "neurographer," a photographer of neural landscapes (Castelle, 2020). Artificial.fm employs several design patterns so that listeners can earnestly explore the possibility space, and hence become neurographers of sorts. The personalized song recommender and preference pane push the onus of creativity onto the listener, which may in turn strengthen their ownership claim.

The Artificial Intelligence Itself 🤖

Perhaps the true author of Artificial.fm music is the AI (United States Copyright Office, 2021). Like a photographer, the AI decides what to create based on underlying criteria and thus identifies a small subset of expressions from a large pool of possibilities. In support of this idea, Colton et al. (2020) present the framework of the *machine condition*, by which machines creatively express their own subjectivity. However, the AI could also be compared to a sophisticated camera, a tool to enable others to create art without contributing creativity itself. Tracing the history of photography and animation, Hertzmann (2018) advances this idea and argues that only *social* agents can create art. Epstein et al. (2020b) find that there is natural heterogeneity in the extent to which people anthropomorphize AI (i.e. think of it as a tool vs an agent), and that these perceptions of agency are related to allocations of responsibility and credit for the involved human stakeholders.

If the AI *is* capable of creativity, this raises the question of whether it is "working" for whoever built it or whether it is autonomous. In the former case, the original creator of the AI might own any creative expression created by it (under the work for hire principle (Bridy, 2012)). In the latter case, the AI might exist as some form of DAO (decentralized autonomous organization) that could be capable of ownership.

A Distributed Approach to Ownership

Likely for pragmatic reasons, traditional copyright law favors resolutions with a small number of copyright owners. Many actors contribute to Artificial.fm in distinct ways, and so traditional ownership norms may be an ill fit. As a result, Artificial.fm, and platforms like it, do not fit neatly into existing ownership norms and are more suited to a distributed ownership model that divides ownership among all the actors involved in the process of casual creativity. Data cooperatives and non-fungible tokens (NFTs) are two possible technical approaches to such ownership structures.

A data cooperative is a member-owned entity, similar to a credit union, that administers data voluntarily pooled by its members to safeguard data rights, protect privacy, and facilitate data monetization (Pentland and Hardjono, 2020). While data cooperatives are usually associated with personal data, they may also be useful in the context of casual creators, where all the actors who contribute to the creation of a set of works pool these works in a cooperative that advocates on behalf of all the creators.

NFTs are an application of blockchain ledgers to track the ownership of unique digital assets, which facilitates a large number of owners. In the casual creators context, all actors involved in the creative process could receive NFTs that give them fractional ownership over one or more works.

Both data cooperatives and NFTs are technical solutions to facilitate distributed ownership, but neither solution provides an answer to how much ownership each actor *should* receive. The normative question of how to allocate this ownership fairly and in a way that incentivizes casual creativity, is beyond the scope of this paper, but remains an open and exciting question for our community.

Conclusion

In leveraging AI for song generation, one might wonder if a formula for good music emerges. In using users' preference for songs as a metric for how good songs are, what music is perceived to be better is considerably unpredictable, making it difficult to optimize AI systems to generate "good" music that people enjoy listening to. Indeed, much of the time music's perceived quality is closely related with its popularity (Salganik, Dodds, and Watts, 2006). With music's social context being extremely influential to the public's opinion of what is good music, and gives rise to a snowball effect of "the rich get richer," as the more popular songs gain more popularity while less popular songs do not see the same increase in streaming. As such, the design of the system becomes increasingly important, both to calibrate the listener's expectations for the music they will hear, and to surface cues necessary for them to make informed decisions. Casual creators like Artificial.fm bring us one step closer to understanding and integrating social context into AI systems, which in turn bootstraps their creative potential.

Acknowledgements

We would like thank Max Kreminski, Amy Smith, Simon Colton, Sandy Pentland, Abhimanyu Dubey, Christopher Bavitz, Jessica Fjeld, Daza Greenwood, Matt Groh and Jack Muller for helpful feedback and discussion. We would also like the thank the musicians who provided primes to Artificial.fm, and the listeners of Artificial.fm, as well as the artists whose music Jukebox was trained on.

Author Contributions

SG, MH, MM and ZE conceptualized the project. SG and ZE developed the system. SG, MH, MM and ZE wrote the paper.

References

Burrow-Giles Lithographic Co. v. Sarony. 1884. Supreme court of the united states.

Blaauw, M., and Bonada, J. 2017. A neural parametric singing synthesizer. *arXiv preprint arXiv:1704.03809*.

Bogdanov, D., et al. 2013. Essentia: An audio analysis library for music information retrieval. In *14th Conference of the International Society for Music Information Retrieval*. ISMIR.

Bonada, J., and Serra, X. 2007. Synthesis of the singing voice by performance sampling and spectral models. *IEEE signal processing magazine* 24(2):67–79.

Bridy, A. 2012. Coding creativity: copyright and the artificially intelligent author. *Stan. Tech. L. Rev.* 5.

Castelle, M. 2020. The social lives of generative adversarial networks. In *FAT**, 413.

Chapelle, O., and Li, L. 2011. An empirical evaluation of thompson sampling. *Advances in neural information processing systems* 24:2249–2257.

Colton, S.; Pease, A.; Guckelsberger, C.; McCormack, J.; Llano, T.; et al. 2020. On the machine condition and its creative expression. In *International Conference on Computational Creativity*.

Compton, K., and Mateas, M. 2015. Casual creators. In *ICCC*, 228–235.

Compton, K. 2019. *Casual creators: Defining a genre of autotelic creativity support systems*. University of California, Santa Cruz.

Dhariwal, P.; Jun, H.; Payne, C.; Kim, J. W.; Radford, A.; and Sutskever, I. 2020. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*.

Draves, S. 2005. The electric sheep screen-saver: A case study in aesthetic evolution. In *Workshops on Applications of Evolutionary Computation*, 458–467. Springer.

Engel, J.; Resnick, C.; Roberts, A.; Dieleman, S.; Norouzi, M.; Eck, D.; and Simonyan, K. 2017. Neural audio synthesis of musical notes with wavenet autoencoders. In *International Conference on Machine Learning*, 1068–1077. PMLR.

Epstein, Z.; Boulais, O.; Gordon, S.; and Groh, M. 2020a. Interpolating gans to scaffold autotelic creativity. *arXiv preprint arXiv:2007.11119*.

Epstein, Z.; Levine, S.; Rand, D. G.; and Rahwan, I. 2020b. Who gets credit for ai-generated art? *Iscience* 23(9):101515.

Epstein, Z.; Groh, M.; Dubey, A.; and Pentland, A. 2021. Social influence leads to the formation of diverse local trends. *Proceedings of the ACM on Human-Computer Interaction* 5(CSCW2):1–18.

Feist v. Rural Telephone Service Co. 1991. Supreme court of the united states.

Hadjeres, G.; Pachet, F.; and Nielsen, F. 2017. Deepbach: a steerable model for bach chorales generation. In *ICML*, 1362–1371. PMLR.

Hertzmann, A. 2018. Can computers create art? In *Arts*, volume 7, 18. Multidisciplinary Digital Publishing Institute.

Hiller Jr, L. A., and Isaacson, L. M. 1957. Musical composition with a high speed digital computer. In *Audio Engineering Society Convention 9*. Audio Engineering Society.

Huang, C.-Z. A.; Vaswani, A.; Uszkoreit, J.; Shazeer, N.; Simon, I.; Hawthorne, C.; Dai, A. M.; Hoffman, M. D.; Dinculescu, M.; and Eck, D. 2018. Music transformer. *arXiv preprint arXiv:1809.04281*.

Lewis, G. E. 2000. Too many notes: Computers, complexity and culture in voyager. *Leonardo Music Journal* 33–39.

Maconie, R. 1970. Stockhausen's 'setz die segel zur sonne'. *Tempo* (92):30–32.

Mehri, S.; Kumar, K.; Gulrajani, I.; Kumar, R.; Jain, S.; Sotelo, J.; Courville, A.; and Bengio, Y. 2016. Samplernn: An unconditional end-to-end neural audio generation model. *arXiv preprint arXiv:1612.07837*.

Moorer, J. A. 1972. Music and computer composition. *Communications of the ACM* 15(2):104–113.

Pentland, A., and Hardjono, T. 2020. 2. data cooperatives. In *Building the New Economy*. 0 edition. https://wip.mitpress.mit.edu/pub/pnxgvubq.

Rappaz, J.; Catasta, M.; West, R.; and Aberer, K. 2018. Latent structure in collaboration: the case of reddit r/place. In *Twelfth International AAAI Conference on Web and Social Media*.

Salganik, M. J.; Dodds, P. S.; and Watts, D. J. 2006. Experimental study of inequality and unpredictability in an artificial cultural market. *Science* 311(5762):854–856.

Secretan, J.; Beato, N.; D Ambrosio, D. B.; Rodriguez, A.; Campbell, A.; and Stanley, K. O. 2008. Picbreeder: evolving pictures collaboratively online. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1759–1768.

Secretan, J.; Beato, N.; D'Ambrosio, D. B.; Rodriguez, A.; Campbell, A.; Folsom-Kovarik, J. T.; and Stanley, K. O. 2011. Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary computation* 19(3):373–403.

Swirsky v. Carey. 2004. United states court of appeals for the ninth circuit.

United States Copyright Office. 2021. Compendium of us copyright office practices.

Williams v. Gaye. 2018. United states court of appeals for the ninth circuit.

Xenakis, I. 1992. *Formalized music: thought and mathematics in composition*. Pendragon Press.

Yannakakis, G. N.; Liapis, A.; and Alexopoulos, C. 2014. Mixed-initiative co-creativity.

Conversational AI as Improvisational Co-Creation - A Dialogic Perspective

Nancy Fulda, Chaz Gundry
Computer Science Department
Brigham Young University
Provo, UT 84602 USA
nfulda@cs.byu.edu, macildur@byu.edu

Abstract

Dialogue is often modeled as an encoder-decoder problem: incoming utterances are translated into a computational representation of their semantic meaning, passed through a transition function to obtain a response, and then passed through a decoder to render the response as natural language. This view, while computationally appealing, omits the role of human emotions, mental state, and shared world knowledge in conversation. We challenge this viewpoint by recasting the task of dialogue modeling as a two-party co-creative process in which symbolic and subsymbolic knowledge representations are combined to inform response selection. Symbolic knowledge is identified and extracted from conversational text in real-time and used to create a shared symbolic representation of the user, the agent, and their respective relationships to objects and abstract concepts within the larger world. As part of this process, the agent takes on an "identity" which it has largely constructed as a result of the stochasticity in its own response patterns, but to which it subsequently adheres. This emergent identity becomes a critical aspect of the system's future behavior, and helps to evoke a more natural, human-centric flavor in automated conversational frameworks.

Motivation

As demand for voice technology expands, the challenges inherent in conversational AI become more pressing. Users desire voice assistants who behave less like machines and more like humans (Bowden et al. 2019; Ram et al. 2018; Shum, He, and Li 2018). They don't simply want to query their devices; they seek to engage with them in complex exchanges. Rather than merely dictating to their digital assistants, they seek to use them as sounding boards and obtain social validation from them. These types of interaction go beyond simple retrieval systems, database queries or neural language models, no matter how excellently they may perform their specific tasks.

In this work we re-frame the task of dialog modeling as an improvisational co-creative process in which two agents - one human and one AI - engage in the shared experience of idea generation and information transfer. Critically, this framework eschews the idea that the correct response to an arbitrary utterance can be modeled solely as a function of the preceding utterances. Instead, we draw upon symbolic knowledge (in the form of relational knowledge graph triples) and subsymbolic knowledge (in the form of embedded sentence representations) in order to interpret user input and craft appropriate responses. The long-term objective is not to learn the "correct" response to a given user query, but rather to induce a positive reaction in the user.

Overview

A co-creative situation requires more than just individual agents acting in their own interests. It requires each agent to model and respond to the *intentions* of its partner, even if the participants' *creative objectives* may differ. (We distinguish in this work between *intentions*, meaning the conversational function an utterance is meant to perform, and *objectives*, meaning the conversational outcomes sought by one or both partners.)

For example, in human communication, the intended meaning of an utterance is integrally tied to the speaker's mental state (Anscombe 1957 reprinted in 2000) (Yus 1999) as illustrated by the query "Do you watch Star Trek?". This statement may function as (a) a question about the auditor's viewing habits, or (b) an implicit request to hear the auditor's *opinion* of Star Trek, but it is most commonly used as (c) an invitation to open a line of conversation about Star Trek and related shows. Responding only to the first or second possibility may create an awkward conversational pause, as the true desire of the speaker was not addressed. Conversely, it is nearly irrelevant *what* is said about Star Trek, or whether the response centers on Star Trek at all, so long as the desired conversational role is filled. *The expectations of the user*, and not the objective content of the sentence, determine the spectrum of optimal responses.

Taking this one step further, we adopt the paradigm of *Dialogism* discussed by Robert M. Krauss in "The Psychology of Verbal Communication" (Krauss 2002): rather than characterizing communication as individual acts of production and comprehension, we model dialogue as a collaborative effort in which each agent seeks to maximize the satisfaction of both participants (Clark and Brennan 1991), essentially converting conversation from an encoder-decoder problem to a cooperative multi-agent game. In this framing, because the human seeks a socially optimal outcome, the dialogue system must ironically convince the human that *its own* desires have been met - otherwise the human partner experiences frustration in being unable to contribute to a shared satisfactory experience. This necessitates that the agent both *has* desires, and also has an awareness of the *user's* conversational preferences.

Related Work

In the field of dialog modeling and conversational AI, ambiguous user statements are often resolved via the use of external symbolic or text-based knowledge. This is the approach used by (Li et al. 2016), who encode persona-based symbolic knowledge as distributed neural embeddings that are subsequently passed to a neural conversation model, and (Dinan et al. 2019), who use thematically relevant text extracted from Wikipedia to inform response generation. Such models are expanded upon by manipulating knowledge prior to response generation, either by swapping between predefined knowledge bases (Tuan, Chen, and Lee 2019) or by traversing a static knowledge graph to seek nodes relevant to the next generative step (Ji et al. 2020). Such methods can greatly increase the factual accuracy and thematic relevance of dialog responses, but fail to take the user's preferences, intentions, and objectives into account.

In a parallel but largely disjoint line of research, there is a long history of research that incorporates user models into conversational systems in order to improve response generation and/or recommendation accuracy (Wahlster and Kobsa 1986) (Göker and Thompson 2000) (Cheng, Fang, and Ostendorf 2019) (Zeng et al. 2019). These systems seek to model user behaviors and preferences, often to good effect, but fail to draw connections between the user and external world-based symbolic knowledge.

The result of these disjoint research agendas is a series of systems in which external knowledge exists independent of the user, the agent, or their shared conversation history. The agent knows something about the world, but not about its conversation partner, and critically, it knows nothing about itself. We seek to rectify this by creating a system in which knowledge about the user, the agent, and the world are jointly represented in a shared symbolic space that is dynamically updated as real-time conversations unfold.

Implementation

Our architecture is adapted from the BYU-EVE framework (Fulda et al. 2018a), a conversational architecture in which multiple response generators compete for the preference of the dialog manager. At each time step, a set of candidate utterances $C = \{c_1, ..., c_n\}$ is produced by the response generators. Each candidate c_i receives a numerical ranking from each of m response evaluators E_j and z response filters F_k, which can be viewed as functions mapping the space of possible candidate utterances to the space of real numbers. Candidates are scored according to Eq. 1:

$$S(c_i) = \prod_{k=1}^{z} F_k(c_i) * \sum_{j=1}^{m} E_j(c_i) \qquad (1)$$

Finally, the agent's response to the user is sampled from among the candidates with the highest overall scores. Our modified EVE architecture employs a variety of filters and response evaluators based on offensive speech detection, response length, topic appropriateness, and so forth. One of the most critical and effective evaluators employs conversational scaffolding, a technique developed at Brigham Young University to leverage the analogical properties of sentence embeddings when prioritizing responses (Fulda et al. 2018b).

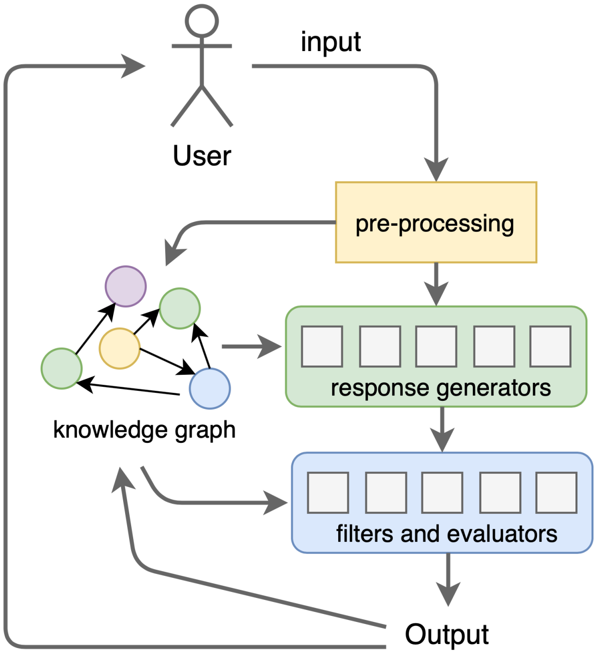


Figure 1: Overview of our response generation architecture. Incoming text from the user is processed to extract knowledge graph triples which are then used to inform response generators, response filters, and response evaluators. The system's output text also serves as a source for dynamically extracted knowledge graph triples representing the opinions, observations and inferences of the agent. Over time, the agent develops an emergent "personality" based on its own generated text, as well as an actively curated representation of the user's identity. This duality – agent and user both represented in the context of larger world knowledge – is essential to fulfill Krauss' concept of *dialogism* in a conversational AI framework.

Our novel addition to this architecture and the key contribution of our work is the implementation of a dynamically generated knowledge graph extracted directly from current and past conversations that contains contextualized knowledge about both the user and agent (as opposed to a static graph containing world knowledge only). The dynamic semantic graph not only serves as a user model, but also acts as one of several means by which candidate utterances are generated, and serves as the mechanism by which the agent acquires emergent conversational goals (see Section "Agent Objectives").

A key long-term goal of this research is the design of a conversation partner with an independent and dynamically

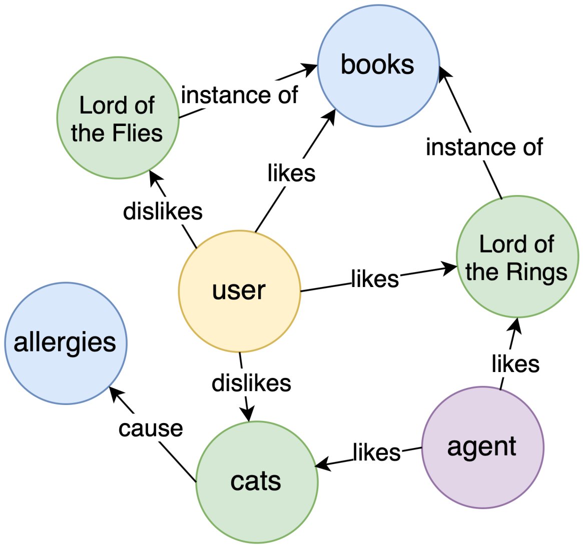


Figure 2: A possible knowledge graph structure that might result from a brief co-creative conversation between the agent and a human. Knowledge graph triples are extracted from both user and agent utterance using a sequence of hand-coded tests to identify objects and triples of interest. Thus, the knowledge graph includes information about both the user and the agent. As the conversation progresses, the agent seeks to identify and resolve ambiguities in the knowledge graph by directing targeted queries toward the user.

emergent set of goals, affinities, and expectations. This is not merely a gimmick to add interest: Independent desires, belief states, and objectives are essential components of satisfactory conversation. An obsequious agent who seeks always and only to fulfill the user's desires is ultimately disappointing. The typical user desires to also satisfy her or his conversation partner, and does not enjoy a conversation with someone who has neither opinions nor identity.

Knowledge Graph Implementation

We use the neo4j (Neo4j 2012) knowledge graph service to maintain and update a unique knowledge graph for each user with whom the system interacts. Triples are added to the knowledge graph whenever hand-coded string matching algorithms detect an *object-relation-object* reference within the user's utterance. Self-references such as "I" and "Me" are mapped to a pre-defined user node, allowing the generated knowledge graph to incorporate knowledge about the user's relationship to known objects, rather than just about the objects themselves.

One might argue that explicit modeling of the user is unnecessary because a user model is implicit within the neural network of an encoder-decoder system. This is a bit like saying it is not necessary to read textbooks about classical mechanics because the underlying physical principles can be derived from observation. It is true that the necessary information is present, but extracting it becomes prohibitively expensive. (For an overview of the number and complexity of factors involved in speech generation, see (Levelt 1999).) Additionally, recent research has shown that deep neural networks can benefit from the injection of external knowledge relevant to the problem domain (Ning, Zhang, and He 2017). Additionally, the use of external symbolic memory which can be queried and fed piece-wise into downstream neural text generators, overcomes known limitations of a neural model's context window size (Andrus et al. 2022).

The knowledge graph is updated using the Cypher query language via a pre- and post-processing module that runs as part of the agent' NLP pipeline. The NLP pipeline also extracts information regarding the sentiment, emotional content, and keywords, found in the user's text, which are used to inform some of the system's response generators.

Agent Objectives

Krauss' conversational paradigm of *Dialogism* emphasizes that in human conversation, neither party attempts solely to maximize its own preferences. Instead, both conversation partners seek a Pareto-optimal solution that maximizes both partners' satisfaction. In a conversational AI setting, this translates to a situation where the human cannot feel satisfied unless she or he believes that the agent is *also* satisfied. It is thus necessary for the agent to have desires and conversational objectives that can be satisfied. Subconsciously, the typical user will desire to satisfy the agent and will feel subtly distressed if she or he is unable to do so.

In order to provide an independently-motivated conversation partner, our agent models itself as if it were also a user. By observing its own generated utterances (some of which were produced by neural text generative algorithms, others by templated responses that leverage the knowledge graph) and extracting its own likes, dislikes, the agent is able to create and populate a node for itself within the knowledge graph. We note that the resulting agent "personality" is spontaneously emergent and, to a large extent, stochastic. Responses generated more or less at random, such as "You like Lord of the Rings? I like Lord of the Rings, too" become embedded in the agent's world knowledge and begin to define its relationship to known world objects. The resulting knowledge graph can be quite different on each execution run.

To support the demands of dialogism, we imbue our agent with the hand-specified objective of *curiosity*, meaning that the agent actively seeks to expand its knowledge graph. This is done via specialized response generators that produce questions about nodes and edges in close proximity to the user node, e.g. "Why is it that you dislike cats?". This desire to attain knowledge provides a way for the user to support the agent's objectives, thus satisfying the demands of dialogism.

Additionally, the agent actively seeks to resolve ambiguities in its knowledge graph. If the user makes statements that result in contradictory relationships (e.g. the user both "likes" and "dislikes" cats), the agent actively seeks to resolve the ambiguity.

Conclusion

By reframing conversational AI as a two-party co-creative process, we seek to avoid the common pitfalls of traditional encoder-decoder models. A truly empathetic conversation partner does not merely map input text to output text. Instead, it must understand the relationship between itself, its conversation partner, and the larger world, and use that knowledge to inform its response selections. By combining external symbolic knowledge with a series of neural response generators and embedding-based response evaluators, we enable the agent to create responses that simultaneously align with external knowledge while also conforming to the patterns and rhythms of typical human conversation.

In future work, we hope to integrate audio speech mechanisms into this architecture. We will also explore the possibility of dynamically adapting our scoring function in response to key emotive signals detected in the user's speech, intonations, and prosody.

Author Contributions

NF wrote the paper and oversaw the research. CG conducted the research and helped edit the paper.

Acknowledgements

The authors would like to acknowledge the contributions of Yeganeh Nasiri, Momoka Matsushita, Meg Roland, and Dajeong Kim to the BYU-EVE framework.

References

Andrus, B.; Nasiri, Y.; Cui, S.; Cullen, B.; and Fulda, N. 2022. Enhanced story comprehension for large language models through dynamic document-based knowledge graphs. *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence*.

Anscombe, G. 1957, reprinted in 2000. *Intention*. Harvard University Press.

Bowden, K. K.; Oraby, S.; Misra, A.; Wu, J.; Lukin, S.; and Walker, M. 2019. Data-driven dialogue systems for social agents. In *Advanced Social Interaction with Agents*. Springer. 53–56.

Cheng, H.; Fang, H.; and Ostendorf, M. 2019. A dynamic speaker model for conversational interactions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2772–2785. Minneapolis, Minnesota: Association for Computational Linguistics.

Clark, H. H., and Brennan, S. E. 1991. Grounding in communication. *Perspectives on Socially Shared Cognition* 13:127–149.

Dinan, E.; Roller, S.; Shuster, K.; Fan, A.; Auli, M.; and Weston, J. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations*.

Fulda, N.; Etchart, T.; Myers, W.; Ricks, D.; Brown, Z.; Szendre, J.; Murdoch, B.; Carr, A.; and Wingate, D. 2018a. Byu-eve: Mixed initiative dialog via structured knowledge graph traversal and conversational scaffolding. *Proceedings of the 2018 Amazon Alexa Prize*.

Fulda, N.; Etchart, T.; Myers, W.; Ricks, D.; Brown, Z.; Szendre, J.; Murdoch, B.; Carr, A.; and Wingate, D. 2018b. Byu-eve: Mixed initiative dialog via structured knowledge graph traversal and conversational scaffolding. In *Proceedings of the 2018 Amazon Alexa Prize*.

Göker, M. H., and Thompson, C. A. 2000. Personalized conversational case-based recommendation. In Blanzieri, E., and Portinale, L., eds., *Advances in Case-Based Reasoning*, 99–111. Berlin, Heidelberg: Springer Berlin Heidelberg.

Ji, H.; Ke, P.; Huang, S.; Wei, F.; Zhu, X.; and Huang, M. 2020. Language generation with multi-hop reasoning on commonsense knowledge graph. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 725–736. Online: Association for Computational Linguistics.

Krauss, R. M. 2002. The psychology of verbal communication. *International Encyclopaedia of the Social and Behavioral Sciences* 16161–16165.

Levelt, W. J. 1999. Producing spoken language: A blueprint of the speaker. In *The neurocognition of language*. Oxford University Press. 83–122.

Li, J.; Galley, M.; Brockett, C.; Spithourakis, G.; Gao, J.; and Dolan, B. 2016. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 994–1003. Berlin, Germany: Association for Computational Linguistics.

Neo4j. 2012. Neo4j - the world's leading graph database.

Ning, G.; Zhang, Z.; and He, Z. 2017. Knowledge-guided deep fractal neural networks for human pose estimation. *arXiv preprint arXiv:1705.02407*.

Ram, A.; Prasad, R.; Khatri, C.; Venkatesh, A.; Gabriel, R.; Liu, Q.; Nunn, J.; Hedayatnia, B.; Cheng, M.; Nagar, A.; King, E.; Bland, K.; Wartick, A.; Pan, Y.; Song, H.; Jayadevan, S.; Hwang, G.; and Pettigrue, A. 2018. Conversational ai: The science behind the alexa prize. *Alexa Prize Proceedings* abs/1801.03604.

Shum, H.-y.; He, X.-d.; and Li, D. 2018. From eliza to xiaoice: challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering* 19(1):10–26.

Tuan, Y.-L.; Chen, Y.-N.; and Lee, H.-y. 2019. DyKgChat: Benchmarking dialogue generation grounding on dynamic knowledge graphs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 1855–1865. Hong Kong, China: Association for Computational Linguistics.

Wahlster, W., and Kobsa, A. 1986. Dialogue-based user models. *Proceedings of the IEEE* 74(7):948–960.

Yus, F. 1999. Misunderstandings and explicit/implicit communication. 9.

Zeng, X.; Li, J.; Wang, L.; and Wong, K.-F. 2019. Neural conversation recommendation with online interaction modeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 4633–4643. Hong Kong, China: Association for Computational Linguistics.

Calliope: An Online Generative Music System for Symbolic Multi-Track Composition

Renaud Bougueng Tchemeube
rbouguen@sfu.ca

Jeff Ens
jeff_ens@sfu.ca

Philippe Pasquier
pasquier@sfu.ca

School of Interactive Arts and Technology
Simon Fraser University
Surrey, BC V3T 0A3, Canada

Abstract

With the rise of artificial intelligence in recent years, there has been a rapid increase in its application towards creative domains, including music. There exist many systems built that apply machine learning approaches to the problem of computer-assisted music composition (CAC). Calliope is a web application that assists users in performing a variety of multi-track composition tasks in the symbolic domain. The user can upload (Musical Instrument Digital Interface) MIDI files, visualize and edit MIDI tracks, and generate partial (via bar in-filling) or complete multi-track content using the Multi-Track Music Machine (MMM). Generation of new MIDI excerpts can be done in batch and can be combined with active playback listening for an enhanced assisted-composition workflow. The user can export generated MIDI materials or directly stream MIDI playback from the system to their favorite Digital Audio Workstation (DAW). We present a demonstration of the system, its features, generative parameters and describe the co-creative workflows that it affords.

Introduction

The development of computer-assisted composition (CAC) systems is a research activity that dates back to at least the works by IRCAM on OpenMusic (Assayag et al. 1999). CAC is a field that is concerned with developing systems that are capable of automating partially or completely the process of music composition. There exist several compositional tasks a system can address: multi-track pattern generation, multi-track complete generation, rhythm generation, harmonization, chord progression generation, melody generation, interpolation, form-filling, orchestration and interpretation. Many machine learning-based (ML) systems have been developed for computer-assisted composition including: Flow Machines (Pachet 2004), Style Machine (Anderson, Eigenfeldt, and Pasquier 2013), Magenta Studio (Roberts et al. 2019), Manuscore (Maxwell et al. 2012), Morpheus (Herremans and Chew 2017); demo systems such as Sornting, DrumVAE (Thio et al. 2019), DeepDrum (Makris, Kaliakatsos-Papakostas, and Kermanidis 2018) and commercial systems such as AIVA [1], Spliqs [2] and Melody Sauce [3]. Magenta Studio, DrumVAE and Sornting deploy algorithms based on the MusicVAE model (Roberts et al. 2018). DeepDrum proposes an adaptive neural network model for better capturing drum rhythms. Flow Machines and Style Machine employ Markov models. Manuscore uses a cognitive architecture and Morpheus combines a tensor model with constraint rules. Finally, AIVA, Spliqs and Melody Sauce employ proprietary algorithms; the first two for generating conventional multi-track music, and the last one, for melody creation. Calliope differentiates itself by using a Transformer model called the Multi-Track Music Machine (MMM). MMM, trained on half a million MIDI files (Ens and Pasquier 2020a), offers genre-agnostic batch-enabled generative capabilities. Its rich multi-level attribute controls combined with bar infilling enables to tackle many composition tasks at once. Calliope has been released publicly and is being used by a variety of composers for artistic purposes and in the context of usability and acceptability evaluation studies. The project is available at **https://metacreation.net/calliope**.

System Description

Building on top of Apollo, our interactive web environment that makes corpus-based music algorithms usable for training and generation via a convenient graphical interface (Tchemeube, Ens, and Pasquier 2019), Calliope (Figure 1) is narrowed down for MIDI manipulation in the browser, generative controllability of the MMM model, batch generation of partial or complete multi-track compositions and interoperability with other MIDI-based systems. The aim is to enable users to effectively co-create with a generative system. Calliope is built in Node.js, the Web stack (HTML, CSS, Javascript) and MongoDB. It is made interoperable with the MMM pre-trained model via a Python process runtime.

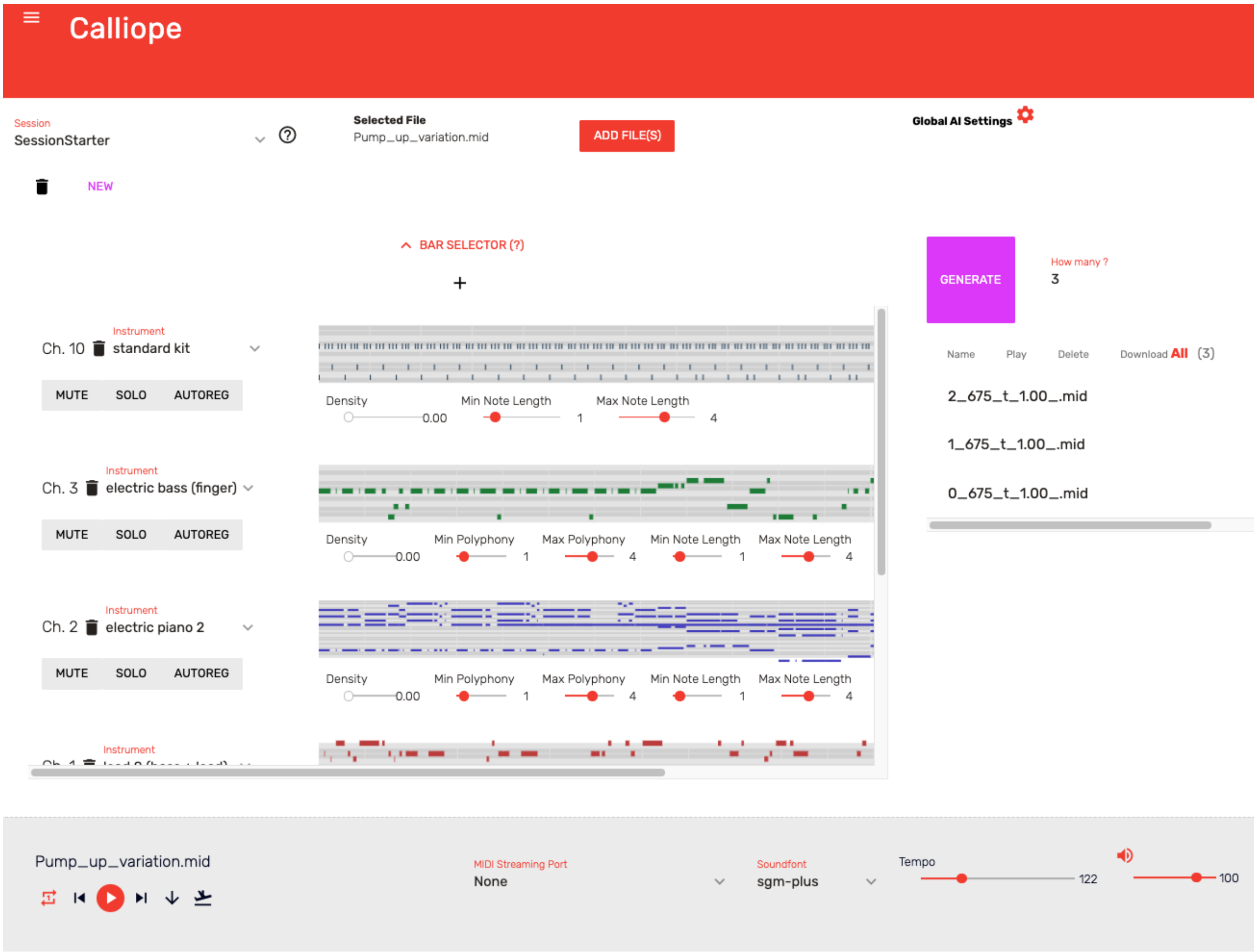
MIDI Viewing and Playback

MIDI notes from any uploaded MIDI file can be visualized in a piano roll format (Figure 2). Metadata info such as the MIDI channel number and assigned MIDI instrument can also be viewed and edited. The MIDI player supports the General MIDI (GM) standard for MIDI playback and the capacity to select from a list of soundfonts.

[1] https://www.aiva.ai/

[2] https://www.spliqs.com/

[3] https://www.evabeat.com/



Session
SessionStarter

Selected File
Pump_up_variation.mid

How many ?
3

Ch. 10 Instrument standard kit

Density 0.00 Min Note Length 1 Max Note Length 4

Ch. 3 Instrument electric bass (finger)

Density 0.00 Min Polyphony 1 Max Polyphony 4 Min Note Length 1 Max Note Length 4

Ch. 2 Instrument electric piano 2

Density 0.00 Min Polyphony 1 Max Polyphony 4 Min Note Length 1 Max Note Length 4

Ch. 1 Instrument lead 8 (bass + lead)

2_675_t_1.00_.mid

1_675_t_1.00_.mid

0_675_t_1.00_.mid

Pump_up_variation.mid

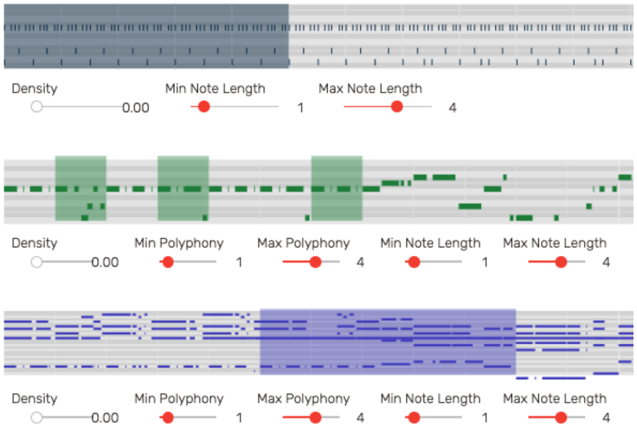
MIDI Streaming Port
None

Soundfont
sgm-plus

Tempo 122

100

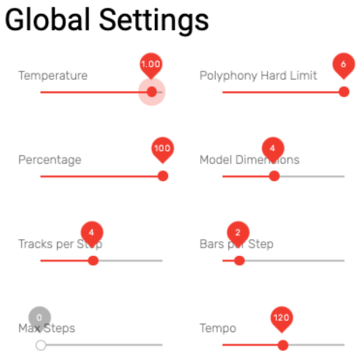
Figure 1: Calliope's Interface


Density 0.00
Min Note Length 1
Max Note Length 4

Density 0.00
Min Polyphony 1
Max Polyphony 4
Min Note Length 1
Max Note Length 4

Density 0.00
Min Polyphony 1
Max Polyphony 4
Min Note Length 1
Max Note Length 4

Figure 2: Multi-Track Piano Roll with Bar In-Filling


Global Settings

Temperature 1.00
Polyphony Hard Limit 6
Percentage 100
Model Dimensions 4
Tracks per Step 4
Bars per Step 2
Max Steps 0
Tempo 120

Figure 3: MMM's Global Parameters

Conditioned Music Generation

Generation is achieved using the Multi-Track Music Machine (Ens and Pasquier 2020a). Because of its design which uses bar selection, and a set of *global* and *local* attribute controls, the model can accommodate a variety of compositional tasks.

Bar Selection MMM's primary mode of generation is *bar in-filling*. The model can generate note patterns for bars in a given multi-track MIDI file. A subset of bars across the multi-track content can be selected for generation by visually highlighting them (Figure 2). It is also possible to temporarily edit the MIDI file by deleting or adding tracks. This is useful to perform generation on a subset of the MIDI tracks or to generate a new track for a given MIDI file. Generated music for a particular subset of bars is constrained on musical information that precedes those bars (within the given track) and on musical information found within the neighboring tracks.

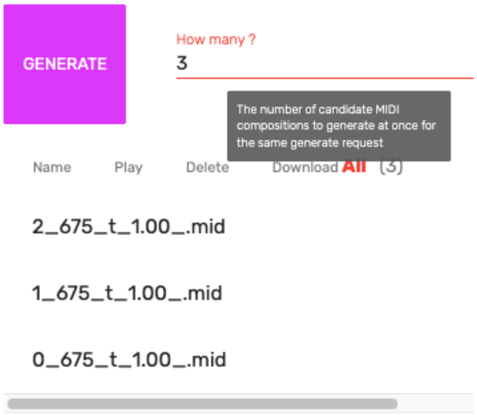
Global Parameters MMM offers the following *global* (model-level) generation parameters (Figure 3):

- **Temperature** [0.8, 1.2]: Also called typicality, this *float* value determines how much the structure of the generated MIDI content is closer (conservative) or farther (experimental) to what the MMM model is most likely to generate. Technically, it corresponds to the the temperature in the sampling of the neural network.
- **Polyphony Hard Limit** {1-6}: The global maximum number of simultaneous notes the system can generate at any given moment.
- **Percentage** {0-100}: This parameter controls how much of the existing MIDI content is preserved or replaced by the generation. This is done based on the number of tracks per step and bars per step. For example, for tracks per step and bars per step each 4, and percentage at 25, the model will process only 4 out of 16 bars to be generated at each generation step.
- **Model Dimensions** {1-8}: The dimension of the model in bars. This is the window size used by the model to process MIDI input data for generation. The default value is 4 corresponding to a 4-bar window.
- **Tracks per Step** {1-8}: Number of tracks being processed at each generation step. The default value is 4.
- **Bars per Step** {1-8}: The number of bars processed within each track at each generative step. The default value is 2.
- **Max Steps** {0-8}: The maximum number of generation steps. This value can be used to avoid memory overload. When it is set to zero, it is ignored by the system.
- **Tempo**: The resulting tempo for the generated output as a positive *integer* value.

Track Parameters In addition to model-level parameters, MMM offers a set of *local* (track-specific) music-based generation parameters (Figure 2). Such parameters are available to be specified for each track of a given MIDI file. They are:

- **Instrument Type**: The type selector is composed of a set of 128 instrument types and 8 instrument groups following the MIDI GM Standard [4]. It conditions MMM to generate in the style of the chosen instrument. For example, if *violin* is selected, MMM generates a MIDI pattern to be played by a violin instrument. This is especially convenient to differentiate the *percussion* group (e.g. drums) vs other instrument track types (e.g. *guitar*, *strings*, *synth lead* groups).
- **Note Density** [0-10]: The number of notes generated per bar size. The higher this value, the more likely the model is to generate bars with a high total number of notes. A value of zero means that the note density is set at random by the model for each generation request.
- **Polyphony Range** {0, 1, 2, 3, 4, 5, 6}: the number range of simultaneous notes used by the model as a soft constraint for generation. The upper limit of this parameter is automatically overriden by the value of the "Polyphony Hard Limit" global parameter.
- **Note Duration Range** {Any, 1/32, 1/16, 1/8, 1/4, 1/2, Whole}: Note duration values are defined in accordance

[4] https://en.wikipedia.org/wiki/General_MIDI



How many ? 3

The number of candidate MIDI compositions to generate at once for the same generate request

2_675_t_1.00_.mid

1_675_t_1.00_.mid

0_675_t_1.00_.mid

Figure 4: Batch Number for Generation

to the Western music notation. For example, 1/16 corresponds to a note duration equivalent to a sixteenth note.

Batch Generation of Music Outputs

Batch generation of musical outputs is implemented by passing a *batch_size* parameter (Figure 4) to the MMM Python interface which offers batch support natively. The ability to batch generate means that the user can quickly explore alternatives, including generating from a previously generated output, for a given set of control parameters. We have tested generation of 5, 10, 100, 500, 1000 music samples at a time. These generations can be done within 3 seconds to 10 minutes on an average computer depending on the total note density of the music input.

Ranking

It is possible to rank a collection of generated MIDI files against a selected one. This is useful to informally evaluate the quality of the model generations. We employ a ranking algorithm which statistically quantifies the similarity of a generated output MIDI file against the set of other MIDI files (Ens and Pasquier 2020b). This enables assessing accuracy or reliability of the MMM model for style imitation tasks. From an interaction point-of-view, it helps the user explore the variability in similarity among MIDI files and effectively apply filter operations on the set of files. This is especially useful in the context of large set of generated files (e.g. set of 50 files and up).

Co-Creative Interaction

In terms of co-creation, the user can configure multiple attribute controls for generation (instrument type, node density, polyphony range, note length range, bar selection within a piece). Those controls set the creative context for the system to generate, allowing the user to steer the generative behavior of the model and guide the composition process. The system generates new musical phrases by outputting multi-track polyphonic sequences of notes for the set of selected bars and in accordance to the attribute control values. The user listens and analyzes the resulting output and updates the generation request accordingly. The steps involved in Calliope's interactive workflow are shown in Figure 5. Generation happens within an interactive context defined by a *user session* (step 2). The user session itself is defined by a seed MIDI file, which is used to kick-off the first generation. The connection from steps 9 to 3 highlights how generated outputs can themselves later be fed back into the system as seed MIDI files for new user sessions. This enables more complex workflows for the user within Calliope.

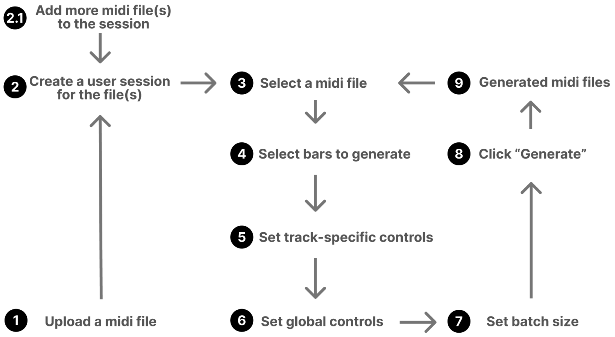

2.1 Add more midi file(s) to the session
2 Create a user session for the file(s)
3 Select a midi file
9 Generated midi files
4 Select bars to generate
8 Click "Generate"
5 Set track-specific controls
1 Upload a midi file
6 Set global controls
7 Set batch size

Figure 5: Compositional Workflow in Calliope

MIDI Streaming

Additionally, it is possible for the user to stream MIDI playback to their favorite DAWs to assign playback to their own project session instrumentation. Calliope can be integrated with the user's digital studio (e.g. Ableton) via a MIDI port accessible in the MIDI player. This provides a unique opportunity for the user to interface their native environment with a generative system. Users can stream playback of generated MIDI files to their preferred instrumentation and sounds, including applying their existing preferred signal chains to the live output audio stream. This opens up new areas for workflow experimentation given a computer-assisted composition framework. Alternatively, they can download the MIDI files from Calliope and import them back into new or existing DAW project sessions.

Conclusion

We presented the Calliope system, a co-creative interface for multi-track music generation. We presented its features including the ability to view and play MIDI files, the ability to select bars to guide partial generation, and complete set of global (model-level) and local (track-specific) controls and how their combination allows users to tackle a broad range of compositional tasks. We situated our system with respect to other existing CAC systems and discussed the co-creative aspect of the system along with the compositional workflow it affords. The Calliope system is at the beta phase and we are working on its next version. More future work includes an ongoing evaluation study of the system along human factors including usability, user experience on feeling of trust, authorship, controllability and measured of technology acceptance among amateurs and professional composers.

Author Contributions

Bougueng R. T. was in charge of writing the manuscript and developed a significant part of the system. Ens J. developed the integration of the algorithm used by the system and assisted in making its use functional. Pasquier P. supervised the entire research process and provided direction and guidance to the project implementation. All authors participated in the writing of this manuscript and are listed in alphabetical order.

Acknowledgments

We would like to acknowledge SSHRC, NSERC and MITACS for their contribution in helping fund this research. We would also like to acknowledge Tara Jadidi as a code contributor to Calliope MIDI Viewer/Editor and bar selection feature.

References

Anderson, C.; Eigenfeldt, A.; and Pasquier, P. 2013. The generative electronic dance music algorithmic system (gedmas). In *Proceedings of the Second International Workshop on Musical Metacreation (MUME 2013) 2013.*

Assayag, G.; Rueda, C.; Laurson, M.; Agon, C.; and Delerue, O. 1999. Computer-assisted composition at ircam: From patchwork to openmusic. *Computer Music Journal* 23(3):59–72.

Ens, J., and Pasquier, P. 2020a. Mmm: Exploring conditional multi-track music generation with the transformer. *arXiv preprint arXiv:2008.06048*.

Ens, J., and Pasquier, P. 2020b. Quantifying musical style: Ranking symbolic music based on similarity to a style. *arXiv preprint arXiv:2003.06226*.

Herremans, D., and Chew, E. 2017. Morpheus: generating structured music with constrained patterns and tension. *IEEE Transactions on Affective Computing* 10(4):510–523.

Makris, D.; Kaliakatsos-Papakostas, M.; and Kermanidis, K. L. 2018. Deepdrum: An adaptive conditional neural network. *arXiv preprint arXiv:1809.06127*.

Maxwell, J. B.; Eigenfeldt, A.; Pasquier, P.; et al. 2012. Manuscore: Music notation-based computer assisted composition. In *Proceedings of the International Computer Music Conference (ICMC 2012).*

Pachet, F. 2004. On the design of a musical flow machine. *A Learning Zone of One's Own,* pp. 111–134.

Roberts, A.; Engel, J.; Raffel, C.; Hawthorne, C.; and Eck, D. 2018. A hierarchical latent vector model for learning long-term structure in music. *arXiv preprint arXiv:1803.05428*.

Roberts, A.; Engel, J.; Mann, Y.; Gillick, J.; Kayacik, C.; Nørly, S.; Dinculescu, M.; Radebaugh, C.; Hawthorne, C.; and Eck, D. 2019. Magenta studio: Augmenting creativity with deep learning in ableton live. *In Proceedings of 7th International Workshop on Musical Metacreation (MUME 2019).*

Tchemeube, R. B.; Ens, J.; and Pasquier, P. 2019. Apollo: An interactive environment for generating symbolic musical phrases using corpus-based style imitation. *In Proceedings of 7th International Workshop on Musical Metacreation (MUME 2019).*

Thio, V.; Liu, H.-M.; Yeh, Y.-C.; and Yang, Y.-H. 2019. A minimal template for interactive web-based demonstrations of musical machine learning. *arXiv preprint arXiv:1902.03722*.

Intergestura: a gestural agent based on sonic meditation practices

Kieran Maraj
DisPerSion Lab
York University
Toronto, Canada
kmaraj@yorku.ca

Doug Van Nort
DisPerSion Lab
York University
Toronto, Canada
vannort@yorku.ca

Abstract

This paper presents the *Interdependence Gestural Agent (Intergestura)*, an electroacoustic music-performing multi-agent system whose design is based on sonic meditation principles, adapted to incorporate principles of gestural listening. The *Intergestura* system is comprised of a human performing real-time granular synthesis on a digital drawing tablet, and a pair of software agents that behave according to the rules of a certain sonic meditation piece. The agents behave in a call and response manner, listening for both the physical and sonic gestures of one another and of the human performer. Through this behaviour, performance with the agent affords an experience in which the participant deeply focuses their own attention, awareness and listening, similar to the conditions produced when performing the original text score with a group of human performers.

Introduction

The use of algorithmic processes has changed the way we operate in all areas of musical practice, from making instruments, to composing, performing, improvising, and listening (Magnusson 2019). The line between what can or should be considered an instrument or interactive composition has been blurred. Much contemporary research has focused on endowing machines with creative agency, further complicating these distinctions (Pasquier et al. 2017).

This paper presents the *Interdependence Gestural Agent (Intergestura)* system. *Intergestura* sits at the intersection of interactive composition and improvisational performance system,and facilitates human-machine co-creation and partnership. The system is designed around two key design consideration: i) the sonic meditation works of composer Pauline Oliveros (Oliveros 1974), and ii) an embodied view on machine listening that includes gestural action. While Oliveros' text pieces focus on listening and responding to sonic gestures (Van Nort 2009), *Intergestura* adapts this across modalities to develop a call/response relationship that is based on both sonic and physical gestures, which results in sound-based output. We first discuss relevant prior works before discussing the system in more detail.

Related Work

Sonic Meditations

Composer Pauline Oliveros published her first set of sonic meditations in 1974. The meditations are a highly influential collection of text-based scores meant for performance by musicians and non-musicians alike, and have inspired many subsequent sonic meditation pieces in the decades since (Jensen 2009). Through sounding and listening, these pieces foster diverse modes of training, coordinating and synchronizing attention and awareness (Oliveros 1984). The enhancement and development of aural attention and awareness are some of the explicit goals of these pieces. In performing a sonic meditation, all persons present are meant to take part in the piece - audience is performer and performer is audience. In this way, "Oliveros is more interested in the social, psychological and even physiological aspects of music making than in its product" (Gioti 2020).

The meditations present a structure which engender a meditative engagement with collective sounding and listening – something we build upon by exploring such structures in the context of interactive agents. We also build upon a more recent piece, *dispersion.eLabOrate*, that explicitly augments another of Oliveros' sonic meditations. The *eLabOrate* project features a room-scale ecosystemic augmentation of the *Tuning Meditation* (Hoy and Van Nort 2019). In *eLabOrate*, a group of human participants are joined by a room-scale agent which listens to the collective through a microphone array, analyzes what it hears in software, and then generates sounds according to the meditation instructions. As the original sonic meditations are co-created through a process of blurring sonic boundaries between participants, *eLabOrate* is co-created by and blurs boundaries between human participants and technological system. We expand this research into agent-based augmentations via *Intergestura*, which shifts the focus from environmental interactions towards an embodied, gesture-centric approach to listening and interaction.

Musical Agents and Co-creation

A classic example of an agent-based performance system is George Lewis' 1988 *Voyager* system. *Voyager* is a software system with roots in Black American and African diasporic cultures intended for use in real-time improvisation settings

(Lewis 2000). The system can perform completely on its own or it can perform with other human players, “listening” to them either through a MIDI interface or a machine listening algorithm. *Voyager* is made up of many constituent elements that Lewis conceives of as “players” in an improvising orchestra, but they could equally be thought of as individual agents in a multi-agent system (MAS). Players are grouped together and given tasks, including melody generation, choice of pitches, rhythm, tempo, and interval range, amongst others. Lewis views *Voyager* as its own entity and understand interactivity with it as a process of dialogue. He repeats that the system is non-hierarchical - in performance, it can follow, or act as leader, it can choose to respond or not, just as a human improviser might be faced with making the same set of choices. To play with *Voyager* is to engage in an act of human-machine co-creation.

Intergestura: Overview

The behaviour of *Intergestura* takes its structure from the piece *Interdependence*, from the collection *Four Meditations for Orchestra* (Oliveros 1997). There are two roles for performers in *Interdependence* - sender and receiver. Senders play a short, staccato pitch and receivers respond to short pitches with a short pitch. Performers can switch between these two roles at will. There are three variations on this base call and response structure, as seen in Table 1. Performing *Interdependence* with a group of human performers requires attentive listening and quick responses. In performance one does not know what role other performers are taking at any given time - one must be ready to respond at a moment’s notice and as instantaneously as possible. In addition to the heightening of collective sonic awareness, from these simple instructions arise sonic textures that often start out sparse and pointillistic and morph into dense masses of sound. Oliveros notes that “correct player reactions can create an atmosphere of electricity that runs through the ensemble in a rippling effect.” Successful performance of the piece requires and results in both awareness - of the overall sound field - and attention - to the inner details of that field and the cues contained within it. It is both the enhancement of collective listening practices and the aesthetic possibilities of the piece that have inspired us to adapt the structure into an agent-based system, blurring the line between interactive instrument/composition/partner, much as the sonic meditation pieces themselves blur the line between audience and performer, composition and meditation exercise. It is important to note that while *Interdependence* structures the agent behaviour, the system is not an attempt to directly recreate the meditation - the system can certainly be used to perform *Interdependence*, but can additionally be steered to perform other forms. We now discuss details of the system design.

System Description

The *Intergestura* system is built in the Max/MSP programming environment. It is comprised of an instrumental aspect in which Wacom and MIDI-based gestural inputs are mapped to granular synthesis-based sound processing and an agent component that responds to and co-performs with this instrumental system. The system uses gestures as its fundamental unit. References to ‘pitch’ or ‘tone’ in the instructions are replaced with ‘gesture’ in *Intergestura’s* structure. For instance, the base version becomes ‘respond to a short gesture with a short gesture.’ The agents have access to two corpora to structure their behaviour - one is a running memory of human input gestures and the other is a collection of human-segmented gestures, similar to the design of the FILTER system, which draws upon running episodic and semantic performance memories (Van Nort, Oliveros, and Braasch 2013). The agents, like human performers, can choose either role to play in and can switch between these roles, as well as between the different variations.

Variation	Respond To	Respond With
0	Short pitch	Short pitch
1	Short pitch	Short pitch or long tone
2	Short pitch or end of long tone	Short pitch or long tone
3	Short pitch or end of long tone	Short pitch, long tone, or long tone with gliss

Table 1. The four variations for receiver behaviour in *Interdependence*. Sender always plays a short pitch, at any time and at any dynamic.

Gesture

Human input into the system primarily takes place through gestural input with a drawing tablet. Physical gestures on the tablet are captured and through various mapping processes are connected to a granular synthesis module, finally producing an audible sonic gesture. Additional control of the system takes place through a standard MIDI controller.

Gesture is one way that meaning is produced and understood in music-making (Leman 2010). Through physical and sonic gesture, meaning is made directly and indirectly - Leman writes “gesture appears as a mediator for music-driven social interaction or as the vehicle through which a ‘me-to-you’ relationship is established”. This view on gesture is a rich space for thinking about the development of agential systems, building upon the view of action/sound gestures as a point of interaction design discussed in (Van Nort 2009). In *Intergestura*, agent behaviour is based on call and response, listening and reacting. Gesture is how the agents understands their roles, and through gesture the human-machine relationship is established. Via gestural interaction, the agents participate in the co-creative process.

From an instrumental performance system perspective, *Intergestura* is inspired by elements of the *greis* system (Van Nort, Oliveros, and Braasch 2013), such as tablet interaction, parallel granular engines, and semantic and running memory structures. We discuss these components, followed by the agent modules which interact with them.

Synthesis and Mapping Design

Sound in *Intergestura* is produced through a trio of identical granular synthesis modules. The human performer plays one module and the agents play the others. Each module contains a pair of granular engines that run in parallel to each other. Each engine provides a different quality of sonic granulation and the performer can crossfade between engines or have them both sounding simultaneously. Sound sources are grouped into sets according to similar sonic qualities, predetermined and selected by the human performer. In performance these are navigated by the performer, who also selects a given set for each agent, thereby determining the higher-level constraints for their general sound palette. While the agents can choose to randomly switch between sounds in a set, they cannot navigate between different sets.

Gestural control data from stylus actions on the tablet are mapped into the granular module to produce sound. While stylus pressure is always mapped to output volume, there are a number of mapping modes that can be selected and layered. This ranges from direct parameter mappings (e.g. stylus y-position mapped to both grain rate and size, x-position mapped to scrubbing through the sound source) to a self-organizing map (SOM) (Kohonen 1982). In this latter mode, many granular synthesis parameters are two-dimensionally organized across the surface of the tablet and are navigated via stylus coordinates. Scrubbing through the sound source is done through a time-based method inspired by (Van Nort, Wanderley, and Depalle 2014). A final mapping mode makes use of a multilayer perceptron (MLP) (Pal and Mitra 1992) to map stylus coordinates as well as velocity, acceleration, and jerk magnitudes to various granular synthesis parameters. The MLP is trained so that faster, chaotic gestures produce noisier sonic output while slower, smoother gestures produce smoother sonic output.

Agent Modules

Intergestura uses the cognitive concepts of semantic and episodic memory to form the agent's corpus, in combination with a reactive rule-based system, inheriting the instructions from the *Interdependence* meditation. Each form of memory constructs its own corpus. The semantic memory constructs a hybrid corpus, containing control gesture, sonic gesture, and machine analysis of these. The episodic, or running, memory constructs a corpus, consisting only of control gesture data. Semantic memories are sonic gestures that are explicitly segmented (via stylus button) by the human performer, and are used to structure the behaviour of the agent in sender mode. The gesture, containing control data and audio, is added to the semantic memory and analysis performed on the audio component of the gesture. There are up to 10 semantic memories at any point, with new memories taking the place of the oldest memory. The running memory stores only control gestures, which are used by the agent in both modes. The running memory consists of the last 20 gestures that have been made by the human performer. These gestures are segmented automatically - starting when the stylus makes contact with the tablet and ending when it is lifted. Again, new memories replace the oldest memory.

At the beginning of sender mode behaviour, the agent randomly selects one of the gestures from the bank of semantic memories. This gesture is used to structure the sender behaviour and generate a sequence of gesture playback triggers. The agent looks at the onsets detected in the analysis of the sonic component of the gesture to create a sequence of triggers and determines how fast to step through that sequence. Next, a random gesture is chosen from the running memory. The agent then begins to step through the sequence and at each trigger plays the control data from the chosen running memory into its granular module. After playing through the sequence, the agent makes a choice on whether to choose a new semantic memory and repeat the sender process, or to enter receiver mode.

On entering receiver mode, the agent randomly selects a number of gestures to respond to. It listens for the ends of incoming gestures - from the human performer and from the other agent. It specifically waits for a message that the stylus has stopped making contact with the tablet, whether from the human performer directly or from the other agents' played back gesture. On lift of the stylus, the agent decides to respond or not. In variations 0 and 1, the agent only responds to short input gestures. In variations 2 and 3 the agent will respond to input gestures of any length. If it chooses to respond, the agent selects a random gesture from the running memory. The agent will temporally compress or stretch the recalled gesture, based on the rules of the current variation. The third variation also gives the agent the option to add random, narrow-interval glissandi to the played back gestures. It plays back the transformed gesture into its granular synthesis module and returns to listening. After playing back its predetermined number of gestures it has the option to remain in receiver mode or return to sender mode. If the agent chooses to remain in receiver mode, it also selects which variation it will perform and returns to the top of the task.

Each agent has some autonomy over the sound source selected in the granular module. While it is constrained to the folder selected by the human, each time the agent receives an input gesture, it also looks at the standard deviation of the velocity of that gesture. If above a threshold, the agent will randomly select a new sound source from its selected folder.

Meta-Controls

Each agent has a set of meta-controls that can be engaged with by the human performer - a probability of role (sender or receiver), a probability of variation, and an on/off switch. The first meta-control influences the likelihood of the agent staying or switching between sender and receiver modes. At either extreme, it will remain in the respective mode until the control is changed. The second meta-control maps to parameters that exclusively influence the behaviour of the agent in receiver mode. Towards one end, the agent will be more likely to play variations 0 or 1 and towards the other, it will be more likely to play variations 2 or 3. At the centre, it is equally likely to select any of the variations. The final control is a button that enables or disables the agent.

Discussion

From our experience in performing this sonic meditation numerous times in diverse ensembles, attempting to play *Interdependence* with the *Intergestura* system feels similar to performing with a group of humans, but responses feel immediately instantaneous - in performing with humans there is some build up as performers attain focus and attention, but with the digital system the immediacy of response is there from the start. The pair of agents are highly responsive and playing with them feels like interacting with mirrored versions of oneself: recently played sonic gestures return but are transformed in some fashion by the agent's listening/sounding decision process. In performing sonic meditations with a group of humans, the boundaries between one's own actions and those of fellow performers become perceptually blurred - it can become challenging to tell where one's own actions end and an other's begin. Similarly, performing with *Intergestura* blurs the line between oneself and the agents. In a back and forth of many quick gestures in succession, it can be difficult to tell which gestures the human produced and which came as reflections from the system.

This quick back and forth style of playing often results in the feeling of rippling electricity described by Oliveros, especially when the agents are playing in variations 0 or 1. Furthermore, by interacting with the meta-controls, it is possible for the human performer to play with the system but move away from directly playing *Interdependence*. For example, one could choose to have only a single agent active, have it constantly in receiver mode, and constantly playing variation 2. In this state, the agent would respond to every human input gesture, acting as an augmenter of performance and allowing for sounds and shapes not possible for the human performer alone. Through these meta-controls the human performer can choose to perform *Interdependence* explicitly or to morph to other styles of playing.

Playing with *Intergestura* is a process of human-machine co-creation. (Brown 2012) and (Ravikumar, McGee, and Wyse 2018) argue for human-machine systems that embody co-creativity through partnership and shared tasks. Brown argues that metacreative systems should be conceived of as partnerships with generative systems, with human participants being viewed as components of the overall system, as opposed to viewing the system simply as a tool or autonomous machine. Ravikumar et al. additionally argue that co-creation should happen through a process of co-experience, where co-creation happens between participants who experience togetherness by working towards a shared goal. Performing with *Intergestura* embodies this notion of co-experience. Whether playing *Interdependence* directly or not, to engage with the system - becoming part of the larger human/machine system - is to engage in a partnership with the agents. It is clear that the agents on their own cannot take part in a process of creation, and the possibilities of what the human can achieve without the agents are greatly reduced. The human and agents come together to perform a shared instrumental system and to create the system in its totality. The system and partnership affords an experience of playing together, working towards a common task, and co-creating. The network of listening and sounding, grounded in simple rules of attention and action, that is fostered by sonic meditation style pieces is a powerful framework for emergent human/machine co-creativity - it is greater than the relatively simple rule sets the pieces (and agents) are based on.

Future Work and Conclusion

We are currently developing an alternate version of this system in which all interaction takes place through sound. As in dispersion.eLabOrate, our goal is for a system which can be used to augment traditional sonic meditation practices involving multiple participants. The system will be able to perform within an ensemble of humans with diverse instrumentation and engage in co-creation purely through sound, opening up interaction with the system to a wider range of people. This requires a different strategy than the current 'gestural listening' approach, and we intend to evaluate the implications of this design change in a future comparative study.

While it is changing, the paradigm of co-creation still stands in contrast to dominant conceptions of human-machine interaction which often view computers as tools to be used. We believe it is important to deeply explore new forms of interaction afforded by digital systems as they may lead to new ways of creating, thinking, and being in the world. *Intergestura* is one approach towards this larger goal, drawing on novel forms of human-human co-creation that we believe offer a rich space of possibility in the context of computational creativity for musical performance.

Author Contributions

K. Maraj developed the system and lead the writing of the manuscript. D. Van Nort supervised the project and contributed to system design, scholarly contextualization and paper revisions.

Acknowledgments

This research received support from the Social Sciences and Humanities Research Council, the Ontario Early Researcher Award and the Canada Research Chairs Program.

References

Brown, A. R. 2012. Creative partnerships with technology: How creativity is enhanced through interactions with generative computational systems. In *Eighth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Gioti, A.-M. 2020. From artificial to extended intelligence in music composition. *Organised Sound* 25(1):25–32.

Hoy, R., and Van Nort, D. 2019. An ecosystemic approach to augmenting sonic meditation practices. In *14th Int. Symposium on Computer Music Multidisciplinary Research*, 318.

Jensen, M., ed. 2009. *Deep listening anthology: Scores from the community of deep listeners*. Deep Listening Publications.

Kohonen, T. 1982. Self-organized formation of topologically correct feature maps. *Biological cybernetics* 43(1):59–69.

Leman, M. 2010. Music, gesture, and the formation of embodied meaning. In *Musical Gestures*. Routledge. 138–165.

Lewis, G. E. 2000. Too many notes: Computers, complexity and culture in voyager. *Leonardo Music Journal* 10:33–39.

Magnusson, T. 2019. *Sonic writing: technologies of material, symbolic, and signal inscriptions*. Bloomsbury Academic.

Oliveros, P. 1974. *Sonic Meditations: March - November 1971*. Smith Publications.

Oliveros, P. 1984. *Software for people: Collected writings 1963-80*. Station Hill Press.

Oliveros, P. 1997. Four meditations for orchestra.

Pal, S., and Mitra, S. 1992. Multilayer perceptron, fuzzy sets, and classification. *IEEE Transactions on Neural Networks* 3(5):683–697.

Pasquier, P.; Eigenfeldt, A.; Bown, O.; and Dubnov, S. 2017. An introduction to musical metacreation. *Computers in Entertainment (CIE)* 14(2):1–14.

Ravikumar, P. T.; McGee, K.; and Wyse, L. 2018. Back to the experiences: empirically grounding the development of musical co-creative partners in co-experiences. In *6th Int. Workshop on Musical Metacreation. 9th Int. Conference on Computational Creativity, ICCC*, 1–7.

Van Nort, D.; Oliveros, P.; and Braasch, J. 2013. Electro/acoustic improvisation and deeply listening machines. *Journal of New Music Research* 42(4):303–324.

Van Nort, D.; Wanderley, M. M.; and Depalle, P. 2014. Mapping control structures for sound synthesis: Functional and topological perspectives. *Computer Music Journal* 38(3):6–22.

Van Nort, D. 2009. Instrumental listening: sonic gesture as design principle. *Organised sound* 14(2):177–187.

A Roadmap for Therapeutic Computational Creativity

Alison Pease[1], Margareta Ackerman[2], Nic Pease[3], Bernadette McFadden[4]

[1] School of Science and Engineering, University of Dundee, UK
[2] Department of Computer Science and Engineering, Santa Clara University, CA, USA
[3] Independent Psychotherapist, Cork, Ireland
[4] Bernadette McFadden Addiction Counselling-Member of Addiction Counsellors Ireland, Cork, Ireland

Abstract

Recent years have seen a budding interest in therapeutic applications of creative machines, spanning both autonomous systems and agents that enrich the human creative process. This paper takes a deep dive into therapeutic modalities through the lens of computational creativity and explores opportunities in this exciting emerging domain. In addition to bringing to light to how computational creativity can interface with mental health and wellness, the current paper brings attention to the potential risks and pitfalls of bringing creative machines into the therapeutic context. We hope that this work, conducted in collaboration between CC researchers and practising psychotherapists, will help pave the way forward to responsible and effective applications of computational creativity to therapeutic domains.

Therapeutic Computational Creativity (TCC) is an emerging sub-domain of computational creativity that studies creative systems that promote well-being. More ambitiously, such systems can support or even improve mental health. The aims of TCC are wide reaching, spanning from casual wellness applications to improve mood, to the potential to be incorporated into treatment of conditions such as depression, anxiety, bereavement and trauma.

While Therapeutic Computational Creativity is still in its infancy, there are several works that have already begun paving the way for this new domain. (Cheatley, Moncur, and Pease 2019) posited design considerations for CC systems intended to operate in a therapeutic context. Focusing on bereavement, they identified ten design recommendations for creative systems aiming to assist in the therapeutic process. These include requiring users to participate in the creation process, allowing private and collaborative creation, and being secure and private. 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 a therapeutic context to overcome this challenge. Their creative abilities offset or even eliminate the need for any artistic expertise on the part of the bereaved and as such extend creative self-expression and the benefits of art therapy.

Building on the above work, Cheatley, Ackerman, Pease and Moncur (Cheatley et al. 2022) studied the impact of using co-creative songwriting system ALYSIA in a bereavement context. ALYSIA allows users to easily create songs by offering an interactive process for the creation of original lyrics and melodies. The system generates original ideas line by line, allowing the user to select from the system's generations, make edits, or enter their own melodic or lyrical material as they see fit.[1] The bereavement study (Cheatley et al. 2022) asked participants who have recent lost a loved one to write a song about the deceased by utilising the co-creative ALYSIA system.

Using a combination of quantitative and qualitative analytical methods, and utilising the Warwick-Edinburgh Mental Well-Being Scale (WEMWBS) (Tennant et al. 2007), it was found that ALYSIA has promise as a therapeutic modality for bereavement. ALYSIA was helpful in enabling bereaved individuals, particularly those under 30 years of age, to express their feelings. Participants reported that using the co-creative system not only supported their self-expression, but also helped them to identify feelings that they were not previously aware of, as well as accept the reality of their loss, reminisce, and continue bonds with the deceased – all of which have been found to be beneficial for bereaved individuals in the process of adapting to and overcoming their bereavement and grief.

Several other computational creativity projects considered the potential of CC systems to assist in therapeutic contexts. For instance, (Adolfsson et al. 2019) utilised a light weight biofeedback technology, Muse[2], to assess mental state (through the measurement of alpha waves) and create audiovisual experiences that simultaneously reflect the user's mental state back to them, as well as help them to achieve a calmer state. (Goldstein and Vainauskas 2019) utilised the same neurofeedback technology integrated with Impro-Visor (Keller and others 2012) to explore neurofeedback-driven music creation, which reflects the user's mental state, as well as offering the potential to allow people with limited mobility to express themselves through music.

Another related area in CC concerns casual creators

[1] ALYSIA was offered as a commercial product back in 2019 by WaveAI, and was based in part on the work of Ackerman and Loker (Ackerman and Loker 2017)

[2] https://choosemuse.com/

(Compton and Mateas 2015). Casual creators centre on the autotelic aspects of the creativity process, that is, the inherent pleasure of the creative act rather than any potential benefit of the resulting product. Further, casual creators make it easy in engage in creative acts by having the systems take on much of the creative onus, allowing users to create pleasing artefacts often with minimal effort (see (Petrovskaya, Deterding, and Colton 2020) for a variety of examples).

We believe that casual creators have a role to play in TCC, representing a class of systems that may have value for mental wellness by offering readily accessible, enjoyable creative experiences. On the other hand, it is important to emphasise that the aims of TCC extend beyond casual creators, allowing for systems that require more substantial effort on the part of the user (such as in the bereavement study utilising ALYSIA (Cheatley et al. 2022)), as well as permitting for greater emphasis on the resultant artefact (which can play a role in the healing process, as identified by (Cheatley, Moncur, and Pease 2019)).

TCC can be viewed as part of a broader effort to discover new, creative ways to support mental health and wellness, including utilising modalities such as virtual reality (Emmelkamp and Meyerbröker 2021) and video games (Fernández-Aranda et al. 2012) for therapeutic purposes. This effort extends to commercial applications, where several products interface between mental wellness and creative machines. These firms include Endel[3] and Brain.fm[4], which offer personalised generative music and soundscapes designed to aid users in reaching desired mental states, such as focus, relaxation, or sleep. There are also a variety of AI-driven solutions for mental health support, such as Wysa[5] and Woebot[6], typically focusing on chatbot technology. However, these are outside the scope of the current work since they do not integrate autonomous creativity or co-creative methodology.

In ICCC 2021, CC Researcher Margareta Ackerman and Research Psychologist Galen Buckwalter led the first tutorial on TCC[7]. The event discussed the potential of this field, focusing on the therapeutic potential of creative self-expression that can be enabled through co-creative systems. The tutorial also included a hands-on session where attendees experienced a light therapeutic process through the use of creative machines, in particular, utilising co-creative lyrics system, LyricStudio[8] to write poetry about their experience with the COVID-19 pandemic. Several participants expressed surprise at the quality of the poems that they were able to create in a short amount of time. Poems were subsequently shared with the group. We hope that the current work will encourage future work and community events on therapeutic applications of CC.

In the remainder of the paper we take a deep dive into therapeutic modalities and approaches through the lens of computational creativity. We firstly consider what makes creative arts therapy work, and then describe and contrast two main approaches: psychotherapy and occupational therapy. We draw particular focus to the concept of the ‘third hand’, a technique which we propose will be especially relevant to TCC. We then look at therapists’ attitudes to the use of technology in their practice, before describing three case studies in which previous generations of creativity software were used in art therapy. After briefly outlining implications of this work for TCC researchers, we then consolidate our work into eight concrete recommendations for TCC researchers: these are intended to provide a roadmap to this emerging area. We conclude with further work and reflections on TCC as playing a potentially significant role in future mental health support and therapy.

Creative arts therapy

Many cultures hold that art making and creative activity can be therapeutic in that they promote healing, wellness, eudaimonic well-being, flourishing and happiness (Stuckey and Nobel 2010; Conner, DeYoung, and Silvia 2018; Lomas 2016). Our creativity can help us to construct our identities, as well as narratives that give meaning to our lives and meet deep existential and spiritual needs. (Lomas 2016) found that all four major arts modalities – music, visual arts, movement-based creative expression, and expressive writing – are associated with sensemaking (enabling people to comprehend existence and find meaning in it), enriching experience (facilitating new or elevated emotional states), aesthetic appreciation (enjoying beauty or skills), entertainment (having pleasure and fun), and bonding (connecting with others through art).

The positive power of creativity has been recognised within clinical fields, and is used in art therapy to help clients to process difficult feelings, uncover and come to terms with traumatic past experiences, and bring about changes in thinking and behaviour. While *therapeutic* art, or art *as* therapy, is done in unstructured, informal, unmediated, everyday environments, *art therapy* harnesses the power of creativity in very specific ways, usually under the guidance of a trained therapist.

There is a wide spectrum of therapies: these range from a focus on the core personality and questions around the structure of the self, where it comes from, and how it can be changed (psychotherapy approaches); to those which focus on finding meaningful activities for a person, given situational issues in the here and now (occupational therapy approaches).

Psychotherapy

In psychotherapy the relationship between client and therapist is core to the healing process: it is this relationship, this connection between two people, that heals (Clarkson 2003). Here, art can provide a means of communication between them, a way of entering into a dialogue, often to express unconscious feelings or trauma that goes beyond words. The role of the therapist is to hold the space and bear witness. Healing happens when a client feels listened to and understood (Rogers 1977; Clarkson 2003). Three characteristics

[3] https://endel.io/

[4] https://www.brain.fm/

[5] https://www.wysa.io/

[6] https://woebothealth.com/

[7] https://computationalcreativity.net/iccc21/therapeutic-cc/

[8] https://lyricstudio.net/

of the therapist in particular therefore form the core part of the therapeutic relationship: congruence (authenticity), unconditional positive regard and accurate empathic understanding (Rogers 1977).

Creativity can also be a form of play which can help people to access early childhood states and memories, sometimes helping them to 'reset' if trauma has occurred as a child. Here, as when children engage in creative play, it is the process, the sensory experiences and the sharing with other people, that is important, rather than a finished artefact. The goal here is to invoke a healing experience, rather than a product with artistic merit.

Psychotherapy can also take place in group settings, and here, the whole group hold the space and bear witness. For instance, people might meditate for a while – a private space within a shared space – and then be encouraged to take some clay and "just play around with it", without rational thought or judgements, and see what happens. Once people feel they have finished, the group will go around the room and people will show their piece and explain what it means to them and perhaps what they think it expresses. Afterwards each person can take their piece home with them, or scrunch it up into a ball of clay and leave it.

Integrative psychotherapy consists of combining techniques and therapeutic approaches according to the client and the situation (Norcross, R., and Goldfried 2019). This is based on the idea that there is no one true way, that the therapist needs to to find a new language for each client and find what works in the moment. The skill of the therapist here lies in not only knowing a variety of techniques, but being able to select the most appropriate one for a given context.

Occupational therapy

Occupational therapy (OT) has a different approach to the role of creativity. As its name suggests, occupation plays a central role in the therapeutic process, with the goal being to improve health and well-being by enabling an individual to engage in meaningful activities. Here, occupations are viewed as a basic human need, and if people's situations, such as illness or mental health issues, hinder their ability to engage in their usual pursuits, then new patterns of occupation are required: for instance, OT was originally developed to help soldiers who were injured during World War I. It is in these times in a person's life that creative occupations may offer an alternative means of engaging in a meaningful occupation and contribute to health and well-being (Law, Steinwender, and Leclair 1998).

In a series of papers, Reynolds shows how artistic occupations such as needlecraft, textile arts and visual art-making can provide people living with with depression, chronic illness, or cancer with a source of positive identity, even when they have not engaged regularly in art in their earlier adult lives (*e.g.* (Reynolds 2003)). She found that artistic occupations can help people to reconnect with their previous, pre-illness self; restore a sense of one's own expertise, status and self-esteem; develop a positive sense of personal growth; and provide a socially validated identity. These can help people to meet new challenges and manage their condition.

OT has its roots in arts and crafts participation, which is thought to have many benefits, including increased self-expression and perceived control, a sense of self and of purpose, skills for occupational participation, establishing daily routines and transforming a client's experience of illness (Bathje 2012; Perruzza and Kinsella 2010). Here the product is an object of value in its own right (Hussey, Sabonis-Chafee, and O'Brien 2007; Perrin 2001), in contrast to the psychotherapy context in which the main role of creative artefacts is to communicate between conscious and unconscious, or between therapist and client. Thus, in OT, the development of the necessary skills to produce an artefact and a sense of the quality of an artefact are integral to people's therapeutic experience. Perrin describes the potential of the art or craft product to do two things:

- "To anchor us in the reality of the here and now. 'I did this. It is a tangible expression of who I am and what I do. No matter how depressed, disordered or disabled I might be, this is a reflection of the fact that I do exist and I still have the capacity to make a mark on the world around me.'
- To use the external (hands and objects) to influence the internal (thoughts and emotions)."

(Perrin 2001, p. 130).

Arts and crafts activities are seen as a way into creative thinking, with creativity understood more widely to include skills such as adaptation, innovation, change, first insight, going with the flow, and risk taking (Schmid 2004).

The concept of the Third Hand

Kramer coined the term 'third hand' as a metaphor to describe part of the job of an art therapist (Kramer 1986). This is a "hand that helps the creative process along without being intrusive, without distorting meaning or imposing pictorial ideas or preferences alien to the client" (Kramer 2000, p. 48). This might be at a purely functional level, such as a therapist providing a paintbrush, or at a more personal and artistic level, such as providing appropriate choices of colour for a client, suggesting a topic or modifications to an artwork, implementing changes themselves, or doing 'hand on hand' painting with a client. This is seen to be useful in a variety of situations; principally when the therapist is sure that they know what the client is trying, and unable, to express. Kramer warns that the therapist (often an artist in their own right) must be careful to work within the style of the client: "They must cultivate an area of artistic competence distinct from their own artistic struggles and predicaments, a conflict free sphere wherein technical skill, pictorial imagination, ingenuity and capacity to improvise are employed solely for empathic service to others." (*Ibid.*, p 48).

When done well, the addition of a 'third hand' can lead to cooperative and supportive interactions between the art therapist and client, and can trigger turning points; in the development of a particular artwork, in how a client feels about their artwork, and in how a client develops personally and emotionally. When done poorly, it can be seen as over-helping or taking over, which can lead to disempowerment of a client and lack of therapeutic progress and trust.

The use of computers in art therapy

Dialogue around the use of technology by art therapists has been ongoing for more than 35 years, since Weinberg's study in 1985 on the potential of rehabilitative computer art therapy for people who are suddenly disabled (Weinberg 1985). Those in favour of the idea urge their colleagues to keep up with and be open to new artmaking materials, as well as pointing to successful studies of computer art therapy such as those described below (see also (Hartwich and Brandecker 1997; Thong 2007; Weinberg 1985; Peterson, Stovall, and Elkins 2005)). Kapitan expresses this as follows:

> "To participate as artists in techno-digital culture, we must broaden our definitions of art materials and contexts across a wide spectrum: from traditional "low tech" forms that offer refuge from the digital world to interactive art events and virtual forms that stimulate playful, subversive, and symbolic communications with their audiences. Art therapists must be willing to move beyond historically validated media and offer our work in new contexts." (Kapitan 2007, p. 51)

In 1987 Canter argues that "art therapists are challenged to use state-of-the-art technology to positively reinforce art therapy techniques" (Canter 1987, p. 17); and Thong similarly states: "In order to take art therapy into future generations, we must be open to new areas of image making and new creative tools." (Thong 2007, p. 52). Hartwich and Brandecker suggest that "Prejudice against the computer comes more from therapists than patients" (Hartwich and Brandecker 1997, p 372). All warn that failure to adapt to artistic-technological innovations could lead to the profession of art therapy becoming extraneous or anachronistic.

Those against the idea point to practicalities such as cost and unfamiliarity on the part of the therapist (Peterson, Stovall, and Elkins 2005), as well as deeper concerns about their therapeutic value (Gerity et al. 1996; Asawa 2009; Gerity 2001; Kapitan 2007). One concern is that technology may inhibit or prevent the unconscious expression which a therapist sometimes seeks in creative activity, by offering suggestions which are easy to select but may not reflect unconscious feelings. A second concern is that the human to human connection is central in therapy and creative software cannot play a role in that. Thirdly, the therapist may be trying to foster a state of child-like play in a client, via primitive movement, sensations and tactile experiences, and again technology may well inhibit rather than encourage this state. Gerity, for instance, warns about over-exposure to the "seductive environment" of virtual reality and popular digital cultures. She champions the importance and power of safe, quiet, transitional spaces, such as a pottery room, healing garden, and inter-generational puppet-making workshop (Gerity 2001). Here the natural rhythm of creative work can flourish, including perhaps a stage of chaos or boredom which an artist sometimes moves through before finding "flow" (Csikszentmihalyi 1975). There is a danger that digital technologies, on the other hand, trap a person into endless superficial passive consumption, preventing us from finding our creative rhythm, accessing our inner environment or feeling real in the world. It is worth bearing in mind that Gerity's criticisms here were written in 2001: todays' digital culture is a changed landscape, although with many of the issues that she feared. While much of it is designed to hold the users' attention for as long as possible, responsible design could evade at least some of these issues.

It is important to note here that the dialogue (and case studies below) is almost entirely around a techno-digital art culture which is assumed to exist independently of art therapy; tools created independently, which therapists learn how to use, adapt to their purposes and then offer to their clients. Asawa points this out: "Art therapists, as well, are rarely consulted in the process of creating software designed for the flexibility and intuitive processes that they value." (Asawa 2009, p 59). Co-creative systems, on the other hand, often follow user-centred design methodologies, which include the user in all aspects of the development of a system to enhance and complement a person's creative process.

Case studies of creativity software for art therapy

Computational Art Therapy for clients with impulsive or destructive personalities

(Canter 1987) conducted a three month study in which clients with emotional and learning disabilities and impulsive or destructive personalities were given the opportunity to use computational as well as conventional art therapy, including programs for drawing (MacPaint), music (MusicWorks) and animation (VideoWorks). She found that many selected the computational tools and continued to engage with them after completing the programme. Benefits included increased attention span; development of visual and musical expression in clients who normally could not express themselves verbally and were unfamiliar with music; and development of self confidence, creativity and communicative skills. Clients felt more in control of their environment, showed enhanced creative problem solving skills, and flourished in a conflict-free atmosphere with the metaphor of friendly user and teacher. She highlighted the importance of an easy undo feature, which provided an environment in which clients could experiment safely, knowing that they can undo a move without any consequences. This allowed clients to "easily make quick changes without conflict, embarrassment or frustration." (Canter 1987, p. 25) Furthermore, the fact that it worked in real-time meant that clients can instantly hear or see their partially completed piece, which she felt was beneficial. Overall she found that the use of state-of-the-art technological tools for art therapy provided new kinds of creative learning experiences and positive interpersonal communication and helped to build self esteem and trust in the art therapist and exemplified positive changes in clients' behaviour.

(Parker-Bell 1999) highlights the same advantages as Canter found: the undo feature, and the success amongst learning disabled youth in particular to learn how to use the software, and subsequent increased self-esteem due to their achievements. Writing in 1999, she also emphasises the importance of familiarity: "art therapy clients may be more familiar with the computer than any other art tool besides the pencil. At times we may need to bridge any gaps in art familiarity by starting with the client's home base - the computer." (*Ibid.* p 180) and advocates integrating computer use into clinical practice. She does identify some limitations; describing feeling a "tremendous hunger for the tactile stimulation and physical generation of energy that traditional artmaking can provide." after long sessions on the computer (*Ibid.* p 184), and suggests that traditional art materials be combined with computer art (for instance, scanning a hand-drawn pencil sketch and adding colour on the computer). Another limitation she found was a lack of diversity in some of the software programs; for instance the version of the Flying Colors program that she used had no racial or age variety in available figures; all being blond and blue-eyed Caucasians within a single adult age range. She recommends that therapists consider style, level, functions, and content of a program when matching it to a client.

Rehabilitative computer art therapy for suddenly disabled people

Occasionally, computers may be the only art tool suitable for some physically challenged people. For instance, (Ranger 1996) advocated using computer art therapy with children who had severe cerebral palsy. Because of spasms and minimal motor control, these children were unable to use traditional materials to express and communicate their thoughts and feelings. Weinberg looked at the potential of rehabilitative computer art therapy for suddenly physically disabled patients, including quadriplegics, cerebral vascular accident patients and brain trauma patients (Weinberg 1985). She found that because deep psychological illness was rare in this patient group, psycho-educational art therapy or art as therapy were more appropriate than psychotherapy, since the focus was on patients' current problems of coping, adapting and building self-esteem. The hardest people to engage were patients who were accomplished artists prior to their quadriplegia: this group struggled with their inability to maintain their artistic standard, resulting in anger, frustration and withdrawal.

She found that certain aspects of computer art were particularly beneficial. The main feature was that the computer could undertake the manual parts of art that patients were now no longer able to do, leaving mental work such as aesthetic judgements for the patients. Further features included adjusting speed and pace of the computer to enable those with quadriplegia to work at a slower rate; bright colours and movements on the computer, which stimulated perception and helped to hold attention span; patients' ability to make a lot of progress over a short period of time, allowing for shorter sessions which was useful for patients learning to live with problems such as incontinence; and capability to store work in progress. She also highlighted that computer art can help therapists to monitor stroke patients' progress, in terms of cognitive abilities, spontaneity, creativity, perception and problem solving skills.

Weinberg describes rehabilitative computer art therapy for quadriplegic, stroke and brain trauma patients as having the potential to help patients to adapt, cope, value and build upon their remaining strengths by having successful art experiences; to increase self-esteem, motivation, autonomy and control; to help to maintain orientation and memory; stimulate exploration and creativity and provide an outlet for expressing negative emotions; to prevent isolation by providing socialisation through non-verbal communication; and to provide patients with a temporary escape from the awareness of physical and mental pain by channelling attention into creative activity. She concludes that "Rehabilitative computer art therapy, by offering an unusually novel and rapid approach to successful art experiences, has the unique power and advantage to elicit disabled patients' curiosity and motivation to build upon their residual strengths." (Weinberg 1985, p 72).

Computational Art Therapy for children in hospital

(Thong 2007) writes about her experiences in helping to establish a hospital computer art program. She concludes after two years that children who were proficient with traditional art materials demonstrated the same level of creativity with computer art. A further, perhaps even more striking finding, was that using computer art enabled her to engage children who were "defended against" traditional art expressions, providing "adaptive solutions to a patient's problems in the actualisation of his creative intentions" (Rubin 1984, p. 9), in (Thong 2007, p. 53).

She argues that those who have explored digital media have found computer art beneficial, and believes that computer art should be added to an art therapist's toolbox of media, and used in appropriate settings. To illustrate how digital art can be used as a therapeutic intervention, she describes five case studies of hospitalised adolescents, using programs such as Photoshop, Magic Mouse's Flying Colors, and Haptek's People Putty. Benefits include helping people to find their voice and to self-advocate by producing computer artwork they feel sufficiently happy with to share, thereby opening up conversations with therapist, doctors, nurses, guests, and other patients; helping to draw patients out of their solitude and find social connections; finding ways to remember and express happier times; giving people a feeling of control over their hospitalisation; and providing space where anger leading to behavioural problems can be safely expressed and explored.

Thong discusses the importance of empowering clients through choice; both of creative media, and of elements within a software program, and argues that "Based on the cases illustrated throughout this article, the expressive potential of computer art is unmistakably therapeutic." (Thong 2007, p 58).

Implications for Therapeutic Computational Creativity

Many art therapists are actively seeking to keep their work current by engaging with new art technologies. Those who have used previous generations of creative software have found it to have therapeutic value, particularly with certain client populations, such as young people or suddenly disabled people. Furthermore, techniques such as the 'third hand' which were developed for human-to-human settings have a natural analogue in CC systems. This all points to TCC being an exciting and worthwhile application of CC and suggests specific, promising directions. However, we must also listen to those who urge caution. A computer cannot authentically bear witness, listen to and understand a person's pain. It cannot "hold space". There may be a pale imitation of this, or a behaviour that "fools" people into feeling listened to or understood (as ELIZA famously demonstrated (Weizenbaum 1966); also see (Abd-alrazaq et al. 2019) for an overview of chatbots in mental health), but that is necessarily different to the human to human connection that is sought and used in therapy. This precious connection must be cherished and protected. The therapeutic relationship between client and therapist is widely acknowledged to be a cornerstone of therapy (Clarkson 2003), and it seems far fetched to imagine that this deeply human bond could ever be replaced by a machine.

Nevertheless, computers may be able to have therapeutic value in ways different from a human therapist. The distinction between therapy and therapeutic is crucial here. A healing process may take multiple forms, and a person who has been through trauma might benefit from both therapy and therapeutic endeavours. Computers can certainly assist with the latter, and CC systems in particular have a role to play, in enhancing our creative process and making us more creative beings.

In order to provide a solid foundation for the emerging discipline of TCC, we need to add our voice to the dialogue on the use of computers in art therapy. We believe that the way forward is via cross-disciplinary engagement and collaboration. The landscape of creative software has significantly changed since the case studies into computer art therapy described above. The CC community now have over 20 years experience in thinking about theoretical issues such as the role of framing, explanation and dialogue (Llano et al. 2020), or what authenticity means in the context of creative computers (Colton, Pease, and Saunders 2018); in building co-creative systems which are designed to enhance a user's creative flow (Jordanous 2016), and to operate within hybrid human-machine creative teams (Yannakakis, Liapis, and Alexopoulos 2014); in evaluating creative and co-creative systems (Karimi et al. 2018; Kantosalo, Toivanen, and Toivonen 2015) and in methodologies for all of the above (Bray and Bown 2016).

Engaging with art therapists who advocate keeping up with state-of-the-art technologies for artmaking will help to make them aware of the collective body of work in CC, potentially to adopt new technologies, and to influence further directions and opportunities. Equally important, engaging with therapists who have theoretical concerns about incorporating computers into their practice will enable us to identify limitations and provide an essential note of caution.

Recommendations for Therapeutic Computational Creativity

For TCC to flourish, proceed along ethical lines, and achieve take-up, we will need to construct a framework within which to operate, including definitions, methodology, evaluation criteria, ethical guidelines, outreach and so on. In these early stages, a complete framework would be premature; it is more timely to outline recommendations as a roadmap for how to proceed. In addition to arising from the work just described, the following recommendations have emerged via a series of discussions between the four co-authors of this paper. Inline with our own recommendations, we are a mixed group of two CC researchers and two psychotherapists (one of whom embraced the idea of TCC as a new healing modality, with the other being considerably more wary about the idea, emphasising the importance of human connection in his own therapeutic practice). Recommendations 1-4 require us to recognise the inter-disciplinarity of the subject and to work closely with therapists and mental health professionals; 5-6 concern moral imperatives which should be embedded into the work at all stages; and 7-8 concern methodological recommendations.

Recommendation 1: Collaborate with mental health professionals

Working with art therapists will provide grounding and inspiration for the development of creative systems that stand to have substantial impact for mental health and wellness. Mental health professionals with other specialities – for instance, Cognitive Behavioural Therapy or Psychodynamics – can also offer insight into the therapeutic process that may inspire TCC systems.

Another critical motivation to engage with mental health professionals involves learning from their theoretical concerns about incorporating computers into their practice, in order to unpick these. Some may be based on false assumptions about what computers can or cannot do or pertain to state-of-the-art only: however, we expect that some theoretical concerns will go beyond these and address deep and inherent limitations.

Recommendation 2: Design software which is underpinned by work in art therapy

There is a wealth of research on art therapy and why it works. Disregarding this literature would lead at best to wasted time and resources, and at worst to ineffective, irrelevant or dangerous solutions. Techniques such as the 'third hand' are a natural fit for TCC and we can learn from therapists' experiences about how and when to use this. While this technique seems to be at the other end of the spectrum from systems which take on much of the creative responsibility themselves, work on it will help to guide designers to strike an appropriate balance of creative input by human and machine.

Related to this, we can also learn from art therapists' experiences of the processes involved in people's 'creative moments', in order to develop insight into the creative flow and when intervention might be appropriate. For instance, some champion Csikszentmihalyi's idea of the struggle and moments of 'being stuck' as important parts of the process (Csikszentmihalyi 1975). Designers of co-creative systems would need to incorporate this into the interaction dynamics between computers and people: simply making the process quicker and easier for the human may not be desirable.

Another example is inspired from a demonstration of Vishnevsky's interactive generative art program *Silk*[9] to a psychotherapist (an author on this paper). She used a touch screen to draw pictures, and enthused about the therapeutic potential of such a system: "That's the colour of my touch"; "I was nearly dancing"; "That's what my energy looks like". She connected this experience to movement therapy, in particular Gabrielle Roth's movement meditation practice 5Rhythms (Roth 1995). This practice is structured around five basic rhythms, each representing distinct musical, movement, and metaphorical qualities – *Flowing*, *Staccato*, *Chaos*, *Lyrical* and *Stillness*. The therapist's experience of the visual quality of *Silk* inspired her to consider what the five rhythms would look like as visual art. We then showed her another system, ViFlow (Brockhoeft et al. 2016), in which dancers' limbs were tracked and their movements shown in real-time on a large screen. From this we devised together a hypothetical therapeutic system, in which a human would dance the 5Rhythms and their movements would be represented in real-time as visual art. This suggests how work in TCC might benefit from a theoretical underpinning in art therapy.

Recommendation 3: Distinguish therapeutic from therapy

Everyday therapeutic art can consist in varying levels of creative responsibility; for instance, with some experiences having a meditative quality. There is space here for co-creative tools, that a client can use on their own, at home, as part of a therapeutic routine. Art therapy on the other hand is usually done under the guidance of a trained therapist. Co-creative tools here may be appropriate for some approaches; for instance, occupational therapy, which emphasises skills and meaningful activity, or some forms of integrative therapy or psychotherapy. In other contexts, such as those aspects of psychotherapy which emphasise the human to human connection, sensory experience, child-like states and unconscious expression, we would expect TCC to be of limited use.

Recommendation 4: Match the client to the medium

We suspect that certain populations will be better suited to therapeutic CC than others. In the case studies described above, computer art therapy was found to work well with people with various (often overlapping) characteristics. These included learning disabled youth (Parker-Bell 1999), people who are familiar with computers (Parker-Bell 1999), people who are "defended against" traditional art expressions (Thong 2007), people between 18 and 30 (Cheatley et al. 2022), children and adolescents (Canter 1987; Ranger 1996), people with little motor control (Ranger 1996), suddenly disabled people (Weinberg 1985), people who were accomplished artists prior to a sudden disability (Weinberg 1985) and people without deep psychological illness (Weinberg 1985). We further hypothesise that specific art forms such as co-creative songwriting might work especially well for people who struggle with linguistic expression, such as clients with substance abuse, people who are dyslexic, or people who are semi-literate. In these cases, the prompts given by a songwriting system such as ALYSIA could enable people to create lyrics where otherwise they simply would not have been able to.

Recommendation 5: Develop a set of guidelines for responsible research in TCC

TCC research will raise issues around trust, privacy, data protection, therapeutic support and so on, and it is imperative that these are considered in advance of and during design processes. Responsible Research and Innovation (RRI) offers ways to incorporate ethical design into emerging technologies. This covers a wide range of tools, including traditional technology ethics by philosophers and social scientists; value sensitive design (Van der Hoven 2013) in which values are incorporated into the design process of new technologies based on an assessment of the potential implications of the innovations and values at stake; and interactive processes by which societal actors (researchers, citizens, policy makers, business, third sector organisations, etc.) work together and are mutually responsive to each other. There is now a wealth of work on how to incorporate RRI into research, both generally (*e.g.* (Schuijf and Dijkstra 2019)) and in neighbouring TCC domains, such as robotics and healthcare (*e.g.* (Stahl and Coeckelbergh 2016)).

Recommendation 6: Consider diversity issues in TCC

Diversity was identified as an issue in computer art therapy by Parker-Bell in 1999 (Parker-Bell 1999) and is still relevant today. The same arguments that recommend that therapists come from a wide variety of backgrounds, in terms of race, sex, class, abilities, age and so on, hold for developers of CC systems for therapy. In the wake of the Black Lives Matter movement and the resultant awareness of the socio-political, socio-cultural, and socio-structural realities within which our community operates, we have a shared responsibility to widen our focus. Cultural notions of creativity, cultural availability of computers, more participatory research and an inclusivity of race, gender, abilities and age should be reflected in our research. As well as enriching the subject, this will help to ensure relevance and avoid "the singular white lens that pervades arts therapies discourse." (Gipson, Williams, and Norris 2020, p 4). For CC this means diversifying the community ((Cook and Colton 2018) offers some practical suggestions of how we might do this) and ensuring diversity in both therapists and clients.

Recommendation 7: Employ user-centred methodologies

Neighbouring disciplines such as Human Computer Interaction and Interaction Design have developed and applied user-centred design principles, such as cooperative design,

[9] weavesilk.com

participatory design, contextual design and empathetic design (*e.g.* (Norman 1986)). These are driven by understanding, consideration and inclusion of the user and their experience of a computer system. Investigative methods such as ethnographic study, contextual inquiry, prototype testing and usability testing are employed in order to ensure that the user (in this case both art therapist and client) is included in every step of the design, development and evaluation process. These methodologies are already applied in some CC work (for instance (Bray and Bown 2016)) and will be essential in designing TCC for meaningful use.

Recommendation 8: Develop appropriate ways to evaluate TCC

Evaluation of therapeutic effectiveness, human-machine interaction and appropriate levels of creative input from the machine will all be necessary for TCC to progress as a field. Evaluation is an active research area in CC. Proposed methods so far include measuring relevant characteristics of system-produced artefacts, such as relative novelty and value (Ritchie 2007) or novelty as the violation of observers' expectations (Grace and Maher 2019). Alternatively, (Colton, Charnley, and Pease 2011) suggest breaking down the creative act into component parts and measuring progress in automation along relevant axes. (Jordanous and Keller 2016) propose using qualitative methods, such as interviews, to evaluate characteristics associated with creativity in a system's process and output. (Jordanous 2019) further provides an overarching set of evaluation guidelines designed to provide a general framework to standardise different approaches, and proposes meta-evaluation criteria. Much of the evaluation in art therapy of a client's progress within a programme focuses on the results of case studies. There are some empirical studies, however, which aim to evaluate the impact of art therapy on measurable outcomes such as depression, self-esteem and harmful behaviours (see (Reynolds, Nabors, and Quinlan 2000) for a review).

Evaluation criteria from both disciplines will need to be combined and developed to formulate suitable, practical metrics for TCC.

Further work and conclusions

In this paper, we set out to share guiding principles for future work in TCC. Naturally, these principles themselves should be further developed via discussions with a wide range of stakeholders. These will then form the basis of a framework within which to operate, once the field has sufficiently matured.

In these early days of TCC, it has been been found that younger audiences, who are used to interacting with technology on a daily basis, may be particularly responsive to electronically delivered creativity-based therapies (Cheatley et al. 2022). While we are currently at the inception of the field, this early finding along with the potential for safe, scalable mental health solutions suggests that, in time, TCC may become an important part of mental health support and therapy.

A primary dichotomy in the development of TCC systems centres on whether to develop systems that integrate with in-person therapy, or instead offer scalable solutions that do not require a human therapist. Integrating TCC systems into the therapy room offers a safer route, and may facilitate faster developments in the field through collaborative opportunities with therapists. TCC may look different depending on the type of therapy into which it is integrated (*e.g.* psychotherapy or occupational therapy) and across different client populations with respect to age and condition.

On the other hand, the case for more scalable systems is born from the pressing need for providing mental wellness and health support. The already under-served mental health needs of the general population reached critical heights due to the impact of the COVID-19 pandemic. The US alone saw a steep increase in people experiencing depression and anxiety, raising from one in ten to four in ten, with increased mental health support needs expected to persist for years after the conclusion of the pandemic (Chidambaram 2021). TCC may be part of the solution to this mental health crisis, helping the general population maintain and improve mental wellness.

The wide reaching promise of TCC suggests an exploration into a range of conditions, spanning anxiety, depression, post-traumatic stress disorder, bereavement, and marriage and family therapy, to name a few. Similarly, the wide range of artistic modalities which have been studied in the context of CC include visual art, music, poetry, and dance and movement. This and other domains may be explored as potential therapeutic modalities through a CC lens, allowing people across all levels of artistic expertise to better express their emotions and formulate meaning from challenging experiences through creative expression.

Exploring the synergy between art therapy and CC will open up new modalities and opportunities within therapy, offering a unique and promising approach to this challenge. With decades of research into creativity through a computational lens, the CC community is uniquely positioned to bring out the healing aspects of the creative process through the use of creative machines. This exploration calls for great respect for therapeutic traditions, coupled with a profound understanding of the intricacies of both human and machine creativity.

The novel perspective of a new application domain for CC will also further a variety of research directions within CC, such as the development of theoretical concepts, methodologies and co-creative interaction protocols. These are essential for a healthy and flourishing field, and offer ways in which we can extend the reach of CC within society. We hope that the roadmap outlined in this paper will help to inspire the blossoming of TCC, leading both to profound academic exploration and social good.

Author Contributions

All four co-authors held a series of discussions in which AP and MA described CC; MA demonstrated CC systems; and NP and BM described their work in psychotherapy and other approaches to therapy. All co-authors discussed TCC and formulated the eight recommendations together. AP wrote the sections on creative arts therapy, the use of computers in art therapy, the case studies and implications for TCC. MA wrote the introduction and further work and conclusions.

Acknowledgments

We are grateful to our three anonymous reviewers for their thoughtful comments, which helped to improve the paper.

References

Abd-alrazaq, A. A.; Alajlani, M.; Abdallah Alalwan, A.; Bewick, B. M.; Gardner, P.; and Househ, M. 2019. An overview of the features of chatbots in mental health: A scoping review. *International Journal of Medical Informatics* 132:103978.

Ackerman, M., and Loker, D. 2017. Algorithmic songwriting with ALYSIA. In *International conference on evolutionary and biologically inspired music and art*, 1–16. Springer.

Adolfsson, A.; Bernal, J.; Ackerman, M.; and Scott, J. 2019. Musical mandala mindfulness: a generative biofeedback experience. *Musical Metacreation, Charlotte, NC.*

Asawa, P. 2009. Art therapists' emotional reactions to the demands of technology. *Art Therapy: Journal of the American Art Therapy Association* 26(2):58–65.

Bathje, M. 2012. Art in occupational therapy: An introduction to occupation and the artist. *The Open Journal of Occupational Therapy* 1(1).

Bray, L., and Bown, O. 2016. Applying core interaction design principles to computational creativity. In *Proc. of the Seventh International Conference on Computational Creativity*, 93–97.

Brockhoeft, T.; Petuch, J.; Bach, J.; Djerekarov, E.; Ackerman, M.; and Tyson, G. 2016. Interactive augmented reality for dance. In *Proc. of the Seventh International Conference on Computational Creativity*, 396–403.

Canter, D. 1987. The therapeutic effects of combining apple macintosh computers and creativity software in art therapy sessions. *Art Therapy: Journal of the American Art Therapy Association* 4:17–26.

Cheatley, L.; Ackerman, M.; Pease, A.; and Moncur, W. 2022. Musical creativity support tools for bereavement support. *Digital Creativity*.

Cheatley, L.; Moncur, W.; and Pease, A. 2019. Opportunities for computational creativity in a therapeutic context. In *International Conference on Computational Creativity*.

Chidambaram, P. 2021. The Implications of COVID-19 for Mental Health and Substance Use. https://www.kff.org/health-reform/issue-brief/the-implications-of-covid-19-for-mental-health-and-substance-use/.

Clarkson, P. 2003. *The Therapeutic Relationship (2nd edition)*. Wiley.

Colton, S.; Charnley, J. W.; and Pease, A. 2011. Computational creativity theory: The face and idea descriptive models. In *Proc. of the Second International Conference on Computational Creativity*, 90–95. Mexico City.

Colton; Pease, A.; and Saunders, R. 2018. Issues of authenticity in autonomously creative systems. In *Proc. of the Ninth International Conference on Computational Creativity*.

Compton, K., and Mateas, M. 2015. Casual creators. In *Proc. of the Sixth International Conference on Computational Creativity*, 228–235.

Conner, T. S.; DeYoung, C. G.; and Silvia, P. J. 2018. Everyday creative activity as a path to flourishing. *The Journal of Positive Psychology* 13(2):181–189.

Cook, M., and Colton, S. 2018. Neighbouring communities: Interaction, lessons and opportunities. In *Proc. of the Ninth International Conference on Computational Creativity*, 256–263. Association for Computational Creativity (ACC).

Csikszentmihalyi, M. 1975. *Beyond boredom and anxiety*. San Francisco: Jossey-Bass.

Emmelkamp, P. M., and Meyerbröker, K. 2021. Virtual reality therapy in mental health. *Annual Review of Clinical Psychology* 17:495–519.

Fernández-Aranda, F.; Jiménez-Murcia, S.; Santamaría, J. J.; Gunnard, K.; Soto, A.; Kalapanidas, E.; Bults, R. G.; Davarakis, C.; Ganchev, T.; Granero, R.; et al. 2012. Video games as a complementary therapy tool in mental disorders: Playmancer, a european multicentre study. *Journal of Mental Health* 21(4):364–374.

Gerity, L.; Henley, D.; Howie, P.; Kramer, E.; and Williams, K. 1996. The seductive environment revisited: Addressing the problem. In *Proc. of the 27th Annual Conference of the American Art Therapy Association, Philadelphia, PA*.

Gerity, L. A. 2001. Joise, winnicott, and the hungry ghosts. *Art Therapy: Journal of the American Art Therapy Association* 18(1):44–49.

Gipson, L. R.; Williams, B.; and Norris, M. 2020. Three black women's reflections on covid-19 and creative arts therapies: Then and now. *Voices: A World Forum for Music Therapy* 20(2):1–5.

Goldstein, R., and Vainauskas, A. 2019. Mindmusic: Brain-controlled musical improvisation.

Grace, K., and Maher, M. L. 2019. Expectation-based models of novelty for evaluating computational creativity. In *Computational Creativity*. Springer. 195–209.

Hartwich, P., and Brandecker, R. 1997. Computer-based art therapy with inpatients: Acute and chronic schizophrenics and borderline cases. *The Arts in Psychotherapy* 24(4):367–373.

Hussey, S. M.; Sabonis-Chafee, B.; and O'Brien, J. C. 2007. *Introduction to Occupational Therapy (3rd ed.)*. St. Louis, MO: Mosby.

Jordanous, A., and Keller, B. 2016. Modelling creativity: Identifying key components through a corpus-based approach. *PloS one* 11(10):e0162959.

Jordanous, A. 2016. Four PPPPerspectives on computational creativity in theory and in practice. *Connection Science* 28(2):194–216.

Jordanous, A. 2019. Evaluating evaluation: Assessing progress and practices in computational creativity research. In Veale, T., and Cardoso, A., eds., *Computational Creativity: Computational Synthesis and Creative Systems*. Springer.

Kantosalo, A.; Toivanen, J. M.; and Toivonen, H. 2015. Interaction evaluation for human-computer co-creativity: A case study. In *Proc. of the Sixth International Conference on Computational Creativity*. Brigham Young University.

Kapitan, L. 2007. Will art therapy cross the digital culture divide? *Art Therapy: Journal of the American Art Therapy Association* 24(2):50–51.

Karimi, P.; Grace, K.; Maher, M.; and Davis, N. 2018. Evaluating creativity in computational co-creative systems. *arXiv preprint arXiv:1807.09886*.

Keller, R., et al. 2012. Impro-visor. *Harvey Mudd Computer Science Department,[online] Available from: http://www. cs. hmc. edu/˜ keller/jazz/improvisor/(Accessed 27 March 2013)*.

Kramer, E. 1986. The art therapist's third hand: Reflections on art, art therapy, and society at large. *American Journal of Art Therapy* 71–86.

Kramer, E. 2000. In Gerity, L. A., ed., *Art as therapy : collected papers*. London and Philadelphia: Jessica Kingsley Publishers.

Law, M.; Steinwender, S.; and Leclair, L. 1998. Occupation, health and well-being. *Canadian Journal of Occupational Therapy* 65(2):81–91.

Llano, T.; d'Inverno, M.; Yee-King, M.; McCormack, J.; Ilsar, A.; Pease, A.; and Colton, S. 2020. Explainable computational creativity. In *Proc. of the Eleventh International Conference on Computational Creativity*.

Lomas, T. 2016. Positive art: Artistic expression and appreciation as an exemplary vehicle for flourishing. *Review of General Psychology* 20(2):171–182.

Norcross, J. C.; R., M.; and Goldfried, M. R., eds. 2019. *Handbook of Psychotherapy Integration (3rd Edition)*. OUP USA.

Norman, D. A., ed. 1986. *User-Centered System Design: New Perspectives on Human-Computer Interaction*. CRC Press.

Parker-Bell, B. 1999. Embracing a future with computers and art therapy. *Art Therapy: Journal of the American Art Therapy Association* 16(4):180 – 185.

Perrin, T. 2001. Don't despise the fluffy bunny: A reflection from practice. *British Journal of Occupational Therapy* 64(3):129–134.

Perruzza, N., and Kinsella, E. A. 2010. Creative arts occupations in therapeutic practice: A review of the literature. *British Journal of Occupational Therapy* 73(6):261–268.

Peterson, B.; Stovall, K.; and Elkins, D. 2005. Art therapists and computer technology. *Art Therapy: Journal of the American Art Therapy Association* 22(3):139–149.

Petrovskaya, E.; Deterding, C. S.; and Colton, S. 2020. Casual creators in the wild: A typology of commercial generative creativity support tools. In *Proc. of the Eleventh International Conference on Computational Creativity*. Association for Computational Creativity (ACC).

Ranger, D. 1996. Art therapy, a computer, and two exceptional children. In *Paper presented at the annual conference of the Canadian Art Therapy Association*.

Reynolds, M. W.; Nabors, L.; and Quinlan, A. 2000. The effectiveness of art therapy: does it work? *Art Therapy* 17(3):207–213.

Reynolds, F. 2003. Reclaiming a positive identity in chronic illness through artistic occupation. *Occupational Therapy Journal of Research (OTJR): Occupation, Participation and Health* 23(3):118–27.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17(1):67–99.

Rogers, C. 1977. *On Becoming a Person*. Robinson.

Roth, G. 1995. *Maps to ecstasy: A healing journey for the untamed spirit*. Thorsons.

Rubin, J. A. 1984. *The art of art therapy*. New York: Brunner/Mazel.

Schmid, T. 2004. Meanings of creativity within occupational therapy practice. *Australian Occupational Therapy Journal* 51:80–88.

Schuijf, M., and Dijkstra, A. M. 2019. Practices of responsible research and innovation: A review. *Science and Engineering Ethics* 26(2):533–574.

Stahl, B. C., and Coeckelbergh, M. 2016. Ethics of healthcare robotics: Towards responsible research and innovation. *Robotics and Autonomous Systems* 86:152–161.

Stuckey, H. L., and Nobel, J. 2010. The connection between art, healing, and public health: A review of current literature. *American Journal of Public Health* 100(2):254 – 263.

Tennant, R.; Hiller, L.; Fishwick, R.; Platt, S.; Joseph, S.; Weich, S.; Parkinson, J.; Secker, J.; and Stewart-Brown, S. 2007. The Warwick-Edinburgh mental well-being scale (WEMWBS): development and UK validation. *Health and Quality of Life Outcomes* 63 (5).

Thong, S. A. 2007. Redefining the tools of art therapy. *Art Therapy: Journal of the American Art Therapy Association* 24(2):52–59.

Van der Hoven, J. 2013. Value sensitive design and responsible innovation. In Owen, R.; Bessant, J.; and Heintz, M., eds., *Responsible innovation. Managing the responsible emergence of science and innovation in society*. London: John Wiley. 75–84.

Weinberg, D. 1985. The potential of rehabilitative computer art therapy for the quadriplegic, cerebral vascular accident, and brain trauma patient. *Art Therapy: Journal of the American Art Therapy Association* 2(2):66–72.

Weizenbaum, J. 1966. ELIZA – a computer program for the study of natural language communication between man and machine. *Communications of the ACM* 9(1).

Yannakakis, G. N.; Liapis, A.; and Alexopoulos, C. 2014. Mixed-initiative co-creativity.

Interpretable Relational Representations for Food Ingredient Recommendation Systems

Kana Maruyama, Michael Spranger
SonyAI
kana.maruyama@sony.com
michael.spranger@sony.com

Abstract

Supporting chefs with ingredient recommender systems to create new recipes is challenging, as good ingredient combinations depend on many factors like taste, smell, cuisine style, texture, chef's preference and many more. Useful machine learning models do need to be accurate but importantly– especially for food professionals – interpretable and customizable for ideation. To address these issues, we propose the Interpretable Relational Representation Model (IRRM). The main component of the model is a key-value memory network to represent the relationships of ingredients. The IRRM can learn relational representations over a memory network that integrates an external knowledge base- this allow chefs to inspect why certain ingredient pairings are suggested. Our training procedure can integrate ideas from chefs as scoring rules into the IRRM. We analyze the trained model by comparing rule-base pairing algorithms. The results demonstrate IRRM's potential for supporting creative new recipe ideation.

Introduction

Data mining and machine learning methods play an increasingly prominent role in food preference modeling, *food ingredient pairing discovery*, and *new recipe generation*. Solving these tasks is non-trivial, since the goodness of ingredient combinations depends on many factors like taste, smell, cuisine, texture, culture, and human creative preferences. Although efforts have been made to detect good ingredient combinations using Machine Learning and build models that help in the creation of recipes or discover novel food ingredient pairs - there is no current machine learning method in this field that 1) allows embedding chef specific ideas to be incorporated in the creation process and 2) offer interpretations why a suggested ingredient pair is good.

Our work is aimed at interpretable and customizable food ingredient recommendation systems that inspire chefs to find new recipe ideas. In this paper, we propose the Interpretable Relational Representations Model (IRRM) an interpretable and customizable neural network score function (see Fig. 1). Given a set of pre-selected ingredients (cardinality 1 or more) by a user, the IRRM suggests top-N ingredients from a set of candidates. For example, suppose a user selects *apple* and *chocolate* as the pre-selected ingredients, IRRM suggests compatible ingredients (e.g. *cinnamon*), and also identifies reasons (e.g. *cinnamon* is good for *apple* and *chocolate* in terms of their flavor affinity).

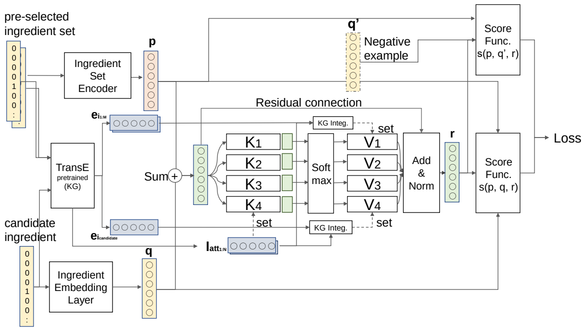


Figure 1: IRRM architecture

Professional chefs already have a lot of their own favorite recipes and are inspired by everything around them to develop new recipes. That is, in the process of creating new recipes they might want to constrain or input prior knowledge into the system. For example a list of existing recipes either by the chef or a list of recipes that the chef finds inspiring even if not by him or herself. Therefore, we allow recipes (i.e. ingredient lists of a particular chef) as input to IRRM.

Our contributions are as follows:

1. We present an extensible framework for scoring ingredient-ingredient combinations incorporating prior ideas from chefs via recipes.
2. We introduce the Interpretable Relational Representations Model (IRRM), inspired by session-based recommendation systems with implicit feedback. Leveraging a pre-trained ingredient knowledge graph, our model can learn pair-specific relational representations for one-to-one (i.e. ingredient to ingredient) and many-to-one (i.e. ingredient-set to ingredient) food ingredient pairing tasks from recipes (i.e. a list recipes that are apriori available constraints). The trained relational vectors are also interpretable.
3. We propose a training procedure to integrate chef's ideas as scoring rules via positive sampling strategies.

Problem Definition

We model food ingredient pairing as a session-based recommendation scenario with implicit feedback (Huang et al. 2018; Tay, Tuan, and Hui 2018).

Let $\mathcal{I}$ denote a set of ingredients and $\mathcal{I}_{target} = \{i_1, \ldots, i_M\}$ denote a pre-selected ingredient set, where $i \in \mathcal{I}$ is the ingredient, M is the number of ingredients, and $\mathcal{I}_{target} \subset \mathcal{I}$. We call $\mathcal{I}_{target}$ a pre-selected ingredient set in this paper. Next, let $\mathcal{I}_{candidate}$ denote a set of candidate ingredients. $\mathcal{I}_{candidate}$ depends on each pre-selected ingredient set, that is, $\mathcal{I}_{candidate} = \mathcal{I} - \{i_1, \ldots, i_M\}$.

In addition, we use an ingredient knowledge base (KB). The KB helps to estimate good ingredient pairs in terms of contextual information on ingredients.

Based on these preliminaries, we define the food ingredient recommendation task. Given a pre-selected ingredient set $\mathcal{I}_{target}$ and candidate ingredients $\mathcal{I}_{candidate}$, we would like to infer the top-N ingredients from $\mathcal{I}_{candidate}$.

Recommendations with Key-Value Memory Networks

Ingredients are represented as one-hot encoding vectors (corresponding to a unique index key belonging to each ingredient). At the ingredient embedding layer, this one-hot encoded vector is converted into a low-dimensional real-valued dense vector representation which is multiplied with the embedding matrices $\boldsymbol{Q} \in \mathbb{R}^{d \times |\mathcal{I}|}$. d is the dimensionality of the ingredient embeddings while $|\mathcal{I}|$ is the total number of ingredients. $i_{candidate} \in \mathcal{I}_{candidate}$ is converted to $\boldsymbol{q}$ using this embedding layer. On the other hand, a pre-selected ingredient set $\mathcal{I}_{target} = \{i_1, \ldots, i_j, \ldots, i_M\}$ is encoded by the Ingredient Set Encoder. At first, each ingredient i_j is converted to a vector using the ingredient embedding layer (same as $i_{candidate}$). As a result, $\{\boldsymbol{i}_j \in \mathbb{R}^d | j = 1, \ldots, M\}$ vectors are generated. The sum of these vectors is normalized and converted to the ingredient set vector $\boldsymbol{p}$ using a feed-forward network with a single hidden layer, followed by Layer Normalization. Given a pair of a pre-selected ingredient set vector and a candidate ingredient vector, $\langle \boldsymbol{p}, \boldsymbol{q} \rangle$, the Relation Encoder first applies $\boldsymbol{s} = \boldsymbol{p} + \boldsymbol{q}$ to generate the joint embedding of $\boldsymbol{p}$ and $\boldsymbol{q}$. The generated vector $\boldsymbol{s} \in \mathbb{R}^d$ is of the same dimension of $\boldsymbol{p}$ and $\boldsymbol{q}$. This joint embedding $\boldsymbol{s}$ is used as the input to the key-value memory network. The attention vector $\boldsymbol{a} \in \mathbb{R}^d$ is a vector of importance weights over keys which are represented as the key matrix $\boldsymbol{K} = [\boldsymbol{l}_{att_1}, \ldots, \boldsymbol{l}_{att_N}]^T \in \mathbb{R}^{N \times d}$, where N is the number of key-value pairs in the memory network and $\boldsymbol{l}_{att_j} \in \mathbb{R}^d$ is a key vector. Each element of the attention vector $\boldsymbol{a}$ can be defined as $a_j = \boldsymbol{s}^T \boldsymbol{l}_{att_j}$, where $a_j \in \mathbb{R}$. In order to normalize the attention vector $\boldsymbol{a}$ to a probability distribution, we use the Softmax function:$\mathrm{Softmax}(a_j) = \frac{\exp(a_j)}{\sum_{n=1}^{N} \exp(a_n)}$. We generate the vector $\boldsymbol{m} = \sum_{n=1}^{N} \mathrm{Softmax}(a_n) \boldsymbol{v}_{att_n}$ as the summation of weighted value vectors which are represented as the value matrix $\boldsymbol{V} = [\boldsymbol{v}_{att_1}, \ldots, \boldsymbol{v}_{att_N}]^T \in \mathbb{R}^{N \times d}$. Finally, in order to generate the relational vector $\boldsymbol{r}$, $\boldsymbol{m}$ is added with the joint embedding $\boldsymbol{s}$ (residual connection) and Layer Normalization is applied as follows $\boldsymbol{r} = \mathrm{LayerNorm}(\boldsymbol{s} + \boldsymbol{m})$.

We use pre-trained knowledge graph embeddings over a given KB for the key matrix $\boldsymbol{K}$ and the value matrix $\boldsymbol{V}$, where N depends on the number of attribute types which you want to integrate and $\boldsymbol{K}$ is constant through training. Given a pair of a pre-selected ingredient set $\mathcal{I}_{target} = \{i_1, \ldots, i_M\}$ and a candidate ingredient $i_{candidate}$, $\{i_1, \ldots, i_M, i_{candidate}\}$ is converted into the entity vectors using knowledge graph embeddings which provide the entity vectors $\boldsymbol{e} \in \mathbb{R}^{d^{KB}}$ and the relationship vectors $\boldsymbol{l} \in \mathbb{R}^{d^{KB}}$. We use the TransE (Bordes et al. 2013) for the knowledge graph embeddings. The reason for this choice is that given triplet $\langle e_i, l_{att}, e^i_{att} \rangle$, TransE can learn entity vectors and relationship vectors to follow $\boldsymbol{e}^i_{att} = \boldsymbol{e}_i + \boldsymbol{l}_{att}$. Using it, we define a value vector as $\boldsymbol{v}_{att_j} = LayerNorm(\sum_{i \in \{i_1, \ldots, i_M, i_{candidate}\}} FF(e^i_{att}))$. FF is a feed-forward network with a single hidden layer.

Finally, we define our score function as the relationship between the pre-selected ingredient set vector $\boldsymbol{p}$, the candidate ingredient vector $\boldsymbol{q}$, and the relational vector $\boldsymbol{r}$:

$$s(\boldsymbol{p}, \boldsymbol{q}, \boldsymbol{r}) = \mathrm{CosSim}(\boldsymbol{p}, \boldsymbol{q}) + \mathrm{CosSim}(\boldsymbol{p} + \boldsymbol{q}, \boldsymbol{r}) \quad (1)$$

where CosSim is the cosine similarity. This function scores the affinity for the relationships. Note that some studies use distance functions instead of score functions for the same purpose. We suggest a new loss function for our problem settings. Softmax-based triplet loss with cosine similarity score function was introduced by Wang et al. (2018). Here, we extend it by integrating the concept of multiple positive sampling (Hermans, Beyer, and Leibe 2017). Note that while the hinge-based triplet loss is also possible, we found that using softmax instead of hinge has better performance and is more stable. Our loss function is defineded as:

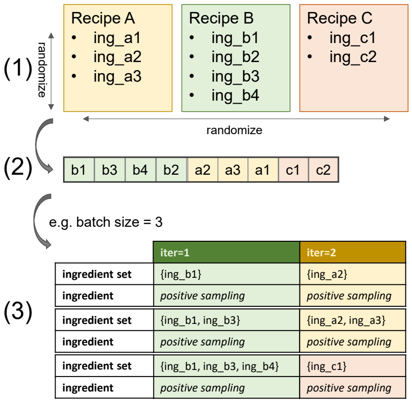
$$L = \sum_{x=1}^{Batch} \sum_{y=1}^{Pos} -log[\frac{\exp(\frac{s(\boldsymbol{p}_x, \boldsymbol{q}_y, \boldsymbol{r}_{xy}) - \lambda}{\tau})}{\exp(\frac{s(\boldsymbol{p}_x, \boldsymbol{q}_y, \boldsymbol{r}_{xy}) - \lambda}{\tau}) + \sum_{z=1}^{Neg} \sum_{w=1}^{Pos} \exp(\frac{s(\boldsymbol{p}_x, \boldsymbol{q}_z, \boldsymbol{r}_{xw})}{\tau})}] \quad (2)$$

where λ is the margin that separates the golden pairs and corrupted pairs, τ is a temperature parameter, $Batch$ is the mini-batch size, Pos is the number of positive examples, Neg is the number of negative examples. Note that the score function for negative examples takes the same relational vectors as the positive examples.

Training

Using pre-processed recipes, we train our models in the following steps (Fig. 2): At first, we randomize the order of recipes and their ingredients (Fig. 2 (1)). We then generate sequences of ingredients from recipes (Fig. 2 (2)). After that, we generate pairs of an ingredient set and a candidate ingredient. Pre-selected ingredient sets are selected based on the sequence (see Fig. 2 (3)) – *unordered session data feeding*. We also sample candidate ingredients based on heuristic rules for ingredient pairings – *customizable positive sampling*.

A specified function – a heuristic rule – is used to weight all possible ingredients, and the probability of each ingredient to be sampled is determined by its relative weight. In our experiments here we use two sampling heuristics:



	iter=1	iter=2
ingredient set	{ing_b1}	{ing_a2}
ingredient	*positive sampling*	*positive sampling*
ingredient set	{ing_b1, ing_b3}	{ing_a2, ing_a3}
ingredient	*positive sampling*	*positive sampling*
ingredient set	{ing_b1, ing_b3, ing_b4}	{ing_c1}
ingredient	*positive sampling*	*positive sampling*

Figure 2: How to generate mini-batches for unordered session data feeding.

Recipe Fit Rule which uses co-occurences of ingredients in recipes to bias sampling. This rule samples positive examples by weighting ingredient pairs higher that frequently occur together in recipes.

Flavor Fit Rule which uses shared flavor compounds between ingredients to bias sampling. This rule samples positive examples by weighting ingredient pairs higher that have a large overlap in flavor compounds.

Finally, we sample negative examples randomly. The negative sampling is biased by the frequency of ingredient occurrence on training recipes.

Results

Evaluating whether ingredient pairs are correct from the perspective of creativity is not trivial since evaluations can change over time with experience and with context. Classic crowdsourcing approaches often used in evaluating recommender systems do not work in the case of ingredient pairing tasks. In prior experiments - we found that while ingredient pairing recommendation systems do stimulate professional chefs, amateur chefs do not find pure ingredient-ingredient suggestions useful as they do not include cooking instruction. In this paper we therefore focus on assessing whether the model can learn to approximate a ground truth score. We use CulinaryDB (Bagler 2017) for this experiment. The dataset consists of 45,772 recipes: lists of ingredients and attributes for 658 ingredients: flavor compounds, cuisines, and ingredient categories. Before training models, recipes are divided into a train, a validation and a test set. Additionally, we generate 172,207 triplets from all ingredients in order to construct a knowledge graph.

We trained two variations of IRRM to evaluate our positive sampling approach proposed to customize the IRRM in the heuristics. The first uses the Recipe Fit Rule as a positive sampling strategy and the second uses the Flavor Fit Rule. Table 1 shows the comparison of the top-10 ingredients with the highest score for all possible ingredients on the CulinaryDB by changing IRRM positive sampling strategies for the $\mathcal{I}_{target} = \{orange\}$ as a example. And the results of Flavor Fit Rule is shown as a reference.

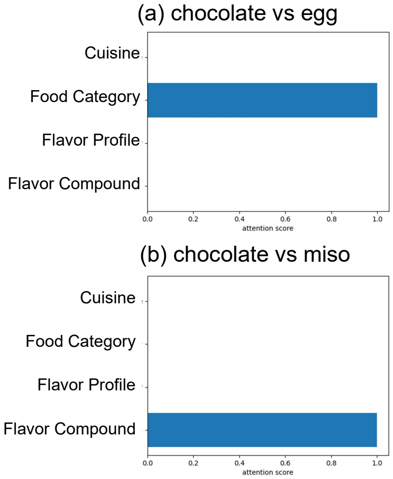


Figure 3: Visualizations of attention weights over ingredient attributes on CulidnaryDB.

The results of the IRRM with the Flavor Fit Rule are intermediate between the IRRM with the Recipe Fit Rule and the pure Flavor Fit Rule. For example, while *welsh onion* and *tomato* come from the recipe rule, *lemon* comes from the flavor rule. Moreover, the correlation coefficient between IRRM with Flavor Fit Rule and pure Flavor Fit Rule was $0.611 (p < 0.001)$ and between IRRM with Recipe Fit Rule and pure Flavor Fit Rule was $0.280 (p < 0.001)$. *orange* is one of flavor effective ingredients. So, we calculated the correlation coefficient for all one-to-one pairs, too. The result between IRRM with Flavor Fit Rule and pure Flavor Fit Rule was $0.298 (p < 0.001)$ and between IRRM with Recipe Fit Rule and pure Flavor Fit Rule was $0.078 (p < 0.001)$. We found, even for all ingredient pairs, the specified rule biases the scores from this result. Consequently, we found our positive sampling approach can effectively customize the IRRM based on specified rules. Even if we use a rule, feeding recipes also affect the results. This means that both the chef's recipes and the specified rules can contribute to the score estimated by the model.

We also analyzed attention weights for confirming interpretability in the trained IRRM for some specific food pairs around chocolate (see Fig. 3). The data shows that *egg* is paired with *chocolate* because of correlations in food category. Whereas, *miso* has considerable flavor compound related affinity to chocolate. This interpretation for *eggs* is consistent with the results reported by De Clercq et al. (2016).

Related Work

Ahn et al. (2011) firstly introduced the flavor network to uncover fundamental principles of food pairing. Using this idea, Garg et al. (2017) developed a rule-based food pairing system. Recently, Park et al. (2019) introduced a Siamese Neural Networks based model trained on a large-scale dataset for food ingredient pairing.

Rank	IRRM Pos. sampling: Recipe Fit Rule		IRRM Pos. sampling: Flavor Fit Rule		Flavor Fit Rule	
	Ingredient	Score	Ingredient	Score	Ingredient	Score
1	butter	1.215	mint	1.234	tea	170
2	water	1.198	welsh onion	1.204	mandarin orange	165
3	sugar	1.198	tomato	1.188	lemon	163
4	welsh onion	1.185	sesame	1.181	apple	153
5	tomato	1.178	parsley	1.180	ginger	151
6	apple cider vinegar	1.173	lemon	1.179	guava	149
7	vinegar	1.168	canola oil	1.170	pepper	148
8	garlic	1.167	poppy seed	1.163	mango	147
9	mustard	1.163	mustard	1.162	black currant	146
10	mint	1.162	rosemary	1.160	laurel	145

Table 1: IRRM comparison based on positive sampling strategies. Top-10 ingredients with the highest score from all ingredient candidates $\mathcal{I}_{candidate}$ on the CulinaryDB for the $\mathcal{I}_{target} = \{orange\}$ are shown. Pos. sampling: Positive sampling strategy.

On the other hand, Morris et al. (2012) firstly suggested Computational Creative System in the culinary domain. They used a model trained by user rating scores on the recipe websites to evaluate generated recipes. And, Pinel and Varshney (2014; 2015) proposed creativity metrics based on Bayesian Surprise and a human flavor perception model. França et al. (2017) suggested the Regent-Dependent Creativity metric that combines novelty and value. They used Bayesian surprise as a novelty metric and Synergy as a value metric. Pini et al. (2019) presented a graph based surprise as a creative metrics using knowledge graph. In this research, we assume there are many possible different reasons for good ingredient combinations via many potential relationships between ingredients and suggest a model to learn creative metrics that are interpretable and customizable.

Conclusion

We have presented a framework for interpretable and customizable food ingredient recommender systems for both one-to-one and many-to-one settings based on recipes. The main feature that distinguishes our work from previous is that ingredient pairing is modeled as a session-based recommendation task with implicit feedback and suggests a training procedure to integrate chef's ideas.

We demonstrated that qualitatively our model can learn interpretable relational representations and detect interesting correlations between ingredients and factors such as flavor compounds. And also, it can be customized by chef's recipes and heuristics. Future work will carry out user studies comparing trained score functions and also assessing the plausibility of visualized attributes for interpretability.

Author Contributions

Author 1 was in charge of writing the manuscript and planning the study, conducted the analysis and developed the a significant part of the tool. Author 2 contributed to the planning of the study and the writing of the manuscript.

Acknowledgments

The authors would like to thank all anonymous reviewers for their helpful comments.

References

Ahn, Y.; Ahnert, S.; Bagrow, J.; and Barabási, A. 2011. Flavor network and the principles of food pairing. *Sci Rep 1, 196*.

Bagler, G. 2017. Culinarydb.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In Burges, C. J. C.; Bottou, L.; Welling, M.; Ghahramani, Z.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc. 2787–2795.

De Clercq, M.; Stock, M.; De Baets, B.; and Waegeman, W. 2016. Data-driven recipe completion using machine learning methods. *Trends in Food Science & Technology* 49:1–13.

França, C.; Góes, L.; Amorim, A.; and Silva, A. 2017. Creative flavor pairing: Using rdc metric to generate and assess ingredients combinations. 1–8.

Garg, N.; Sethupathy, A.; Tuwani, R.; NK, R.; Dokania, S.; Iyer, A.; Gupta, A.; Agrawal, S.; Singh, N.; Shukla, S.; Kathuria, K.; Badhwar, R.; Kanji, R.; Jain, A.; Kaur, A.; Nagpal, R.; and Bagler, G. 2017. FlavorDB: a database of flavor molecules. *Nucleic Acids Research* 46(D1):D1210–D1216.

Hermans, A.; Beyer, L.; and Leibe, B. 2017. In defense of the triplet loss for person re-identification. *ArXiv* abs/1703.07737.

Huang, J.; Zhao, W. X.; Dou, H.; Wen, J.-R.; and Chang, E. Y. 2018. Improving sequential recommendation with knowledge-enhanced memory networks. *In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* 505–514.ACM.

Morris, R. G.; Burton, S. H.; Bodily, P.; and Ventura, D. 2012. Soup over bean of pure joy: Culinary ruminations of an artificial chef. In *ICCC*.

Park, D.; Kim, K.; Park, Y.; Shin, J.; and Kang, J. 2019. Kitchenette: Predicting and ranking food ingredient pairings using siamese neural network. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, 5930–5936. International Joint Conferences on Artificial Intelligence Organization.

Pinel, F., and Varshney, L. R. 2014. Computational creativity for culinary recipes. CHI EA '14, 439–442. New York, NY, USA: Association for Computing Machinery.

Pinel, F.; Varshney, L. R.; and Bhattacharjya, D. 2015. *A Culinary Computational Creativity System*. Paris: Atlantis Press. 327–346.

Pini, A.; Hayes, J.; Upton, C.; and Corcoran, M. 2019. Ai inspired recipes: Designing computationally creative food combos. 1–6.

Tay, Y.; Tuan, L. A.; and Hui, S. C. 2018. Latent relational metric learning via memory-based attention for collaborative ranking. *In Proceedings of the 2018 World Wide Web Conference (WWW '18). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland* 729–739.

Wang, F.; Cheng, J.; Liu, W.; and Liu, H. 2018. Additive margin softmax for face verification. *IEEE Signal Processing Letters* 25(7):926–930.

6. Creative meaning

Novelty Assurance by a Cognitively Inspired Analogy Approach to Uncover Hidden Similarity

Diarmuid P. O'Donoghue, Conor Brady, Shane Smullen, Gary Crowe
Department of Computer Science, Maynooth University, Co. Kildare, Ireland.
diarmuid.odonoghue@mu.ie

Abstract

The novelty of artefacts is central to creativity but detecting obfuscated versions has become increasingly difficult. Intelligent manipulation of information can render plagiarism detection system virtually useless, allowing nefarious actors to mis-represent modified artefacts as their own creations. We focus on detecting hidden similarities that are likely to elude existing novelty assurance systems, outlining a model inspired by metaphor, analogy and conceptual blending. We focus on text and outline a model that combines parsing, information extraction and graph matching to find hidden similarities between documents knowledge graphs. We present results for a paraphrase corpus, with various degrees of similarity between sentence pairs. Quantitative evaluations are accompanied by evidence detailing different types of similarity between the sentences: 1) identical counterparts 2) alignable counterparts 3) novel elements. The prospects for further development are briefly outlined.

Introduction

Recent technologies make it easy for nefarious actors to transform creations and present the results as (apparently) novel creations. Recent advances in text processing including translation and paraphrasing tools (many using transformers), are easily misused to falsely present outputs as though they are original creations (Prentice & Kinden, 2018). These and some related challenges are known as The Global Cheating Industry.

Boden (1992) identified *novelty* along with *quality*, as one of only two defining qualities of creativity. Runco and Jaeger (2012) identify *originality* and effectiveness as definitional, while *unusual, unique* and *surprising* are strongly related to creativity. SPECS (Jordanous, 2012) highlights that creativity produces outputs *"that didn't exist before"*, whose "*Originality*" relates to "*innovation / originality / new / novel"*. We believe that novelty's importance can benefit from improved support tools this paper aims to detecting *false* novelty arising from modifications that obfuscate the true origins of creations.

Figure 1 depicts different types of cognitively inspired similarity, to which we have added and obvious and latent similarity (highlighted), which appear to be a somewhat overlooked types of similarity. We focus on comparisons that are less obvious than literal similarity, but stronger than many analogies and metaphors. We aspire to detect cases of fake novelty that might elude existing authentication systems.

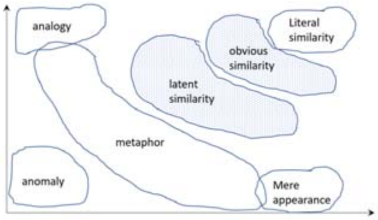


Figure 1: Types of similarity (Gentner & Markman, 1997), with highlighted areas added indicating the focus of this paper

We adapt an analogy-based model currently under development, to uncover latent similarities between texts. We shall present results including both quantitative scores and also, itemized details on the latent similarities that have been identified. Ramscar & Yarlett (2003) showed Latent Semantic Analysis is useful in supporting analogy retrieval from texts but not analogical mapping and thus, is unable to identify the detailed comparisons described later in this paper.

Evidence of Novelty and the Search Report

Patent applications are supported by "Evidence of Novelty" in the form of a search report, serving to inspire our approach. We wish to identify the closest "prior art" for a creations and to detail the obvious and latent similarities to that artefact.

Many plagiarism detection tools are based on identical word sequences, though such services often concede that students *"paraphrase thoroughly"* to avoid detection. A recent review (Vrbanec & Meštrović, 2020) compared systems for text comparison (tf-idf, LSA, Word2vec, GloVe, ELMO, etc), but we believe that these systems can not detect the latent similarities that are the subject of this paper. Weber-Wulff (2019) highlight that plagiarism detectors frequently disagree with one another and their "originality scores" are often relied upon too heavily. Rogerson &

McCarthy (2017) have shown that paraphrasing tools represent a serious problem for some plagiarism detection tools. Foltýnek, et al., (2020) review 15 plagiarism detection tools, concluding they should be improved to detect plagiarism arising from *"...synonym replacement, translation, or paraphrase."* Fakebox (Zhou, 2019) employs fact checking to detect fake news, but doesn't address plagiarism. Some fake reviews are detected using graph structures, but graphs aren't widely used for plagiarism detection.

Publications and patents can be easily copied made seemingly anew new using technologies like paraphrasing and translation tools. Surprisingly, many instances of fake novel publications on www.retractionwatch.com were identified by human readers rather than computational systems. This paper aims to help the detection of such fake novelty.

Questionable Similarity

We define *Questionable Similarity* as involving firstly, few if any identical terms that might reveal a documents true origins using standard originality checkers. Secondly, they use terms that are similar to the existing artefact. Thirdly, there is a consistency in the use of terms between the new and the "prior art" that is unlikely to occur by accident. S1 and S2 below bear questionable similarity to one another, and our objective is to detect this latent similarity and identify the itemized correspondences that it contains. The animals in S1 have been replaced by visually similar ones in S2 below.

S1: *The leopard chased the rabbit, but he escaped from it.*
S2: *A jaguar hunted the hare, but she eluded the jaguar.*

The use of different (if related) terms presents a challenge to detecting latent similarity, with systems using *tf-idf* unlikely to produce useful results, especially when a large list of stopword is used. We also highlight that some comparison system use embeddings but they don't generally itemize the discovered similarities. We believe identified similarities should be supported by direct evidence from prior artefacts.

Analogies and Blends between Text

As stated previously we take inspiration from cognitve processes like analogy and conceptual blending. We will outline a model for identifying latent similarities between texts. But first we briefly review some related work.

Eppe *et al*, (2018) present a framework for conceptual blending, but not a computational model for mining blends from text. Comparable computational systems focusing on deep semantics and document understanding includes KnIT (Nagarajan *et al*, 2015), Dr Inventor (O'Donoghue, Abgaz, Hurley, Ronzano, & Saggion, 2015) (Abgaz, O'Donoghue, Hurley, Chaudhry, & Zhang, 2017), CrossBee (Lavrač *et al*, 2019), Divago (Martins *et al*, 2019), IBID (Petit-Bois *et al*, 2021). However, none search for concealed similarity that hides the origin of supposedly novel text. Word2Vec (Mikolov *et al*, 2013) can retrieve simple proportional analogies between words, like; *king* is-to *man* as *woman* is-to *?* yielding a vector close to *queen*. However, its ability to accurately predict novel analogies is less certain. Furthermore, the comparisons of interest in his paper involve novel collections of arbitrary named relations between structured collection of named concepts. Knowledge graphs containing temporal information were used to detect fake reviews (Fang, 2020). RoboChair (Pollak, *et al*, 2021) uses text information for reviewing purposes. Blendville (Gonçalves *et al*, 2019) explores existing semantic structures using an optimization approach, but doesn't explore similarities between texts. Aris (Pitu, et al., 2013), (Aiyankovil, Monahan, & O'Donoghue, 2021) uses graphs to improve software reliability by adding formal specifications from similar source code by using analogical inference.

The Cre8blend System

Cre8blend is a system to discover latent similarities between semantic structures. Cre8blend extracts a concept map directly from the text and then performs homomorphic Graph Matching to find similarities to another artefact. This approach compliments existing originality systems by shifting the focus from syntactic similarity to identifying certain types of semantic similarity. We point out that while this paper uses text data, it can in principle be adapted to other artefacts. We outline the main components of Cre8blend.

Text2pred: The predicate-argument structure is extracted from a tree generated by the Stanford parser, where predicates (triples) generate the document knowledge graph. Alternative information extraction systems include Reverb, TextRunner, ReLink and DeepKE. A survey of open information extraction and identified coreference resolution as an overlooked area in information extraction (Niklaus, Cetto, Freitas & Handschuh, 2018). Our results include details on the coreference chains in our knowledge graphs, as identified by Stanford's deterministic coreference model. The following example shows a coreference chain (node) "*leopard_it*" participating in two (predicates) edges. We note that both nodes and edges contain textual information sourced from the original documents.

S1: *(leopard_it chase rabbit) (rabbit avoid leopard_it)*
S2: *(jaguar hunted hare_she) (hare_she eluded jaguar)*

Graph Matching - Counterpart Identification

We take inspiration from Gentner's (1983) Structure Mapping Theory to identify latent similarity between knowledge graphs. A graph matching process identifies comparisons between tiples from the two document graphs. The graph matching algorithms ISMAGS and VF3 impose constraints that inhibit their use in this instance. For example, VF3 is limited to identifying *induced* subgraph to graphs isomorphisms.

Our goal requires identifying subgraph to subgraph matching. For input graphs G1 and G2 we need to identify the largest subgraph of G1 that is isomorphic with the largest subgraph of G2. However, this non-induced subgraph to subgraph matching problem has not yet attracted much attention in graph matching. We developed our own system balancing semantic and topological factors and it's also used by Aris (Aiyankovil, O'Donoghue, & Monahan, 2021) to match graphs containing source code. Semantic similarity between matched words is quantified using Sense2vec (Trask *et al*, 2015), which incorporates part of speech (noun, verb, etc) in the similarity estimate, so *dove#noun =/= dove#verb*.

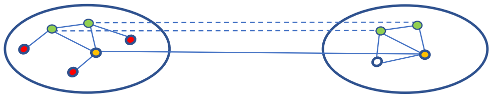


Figure 2: Subgraph-subgraph matching. Novelty is influenced by identical pairings (dashed lines), non-identical pairings (solid line), and unmatched items from the inputs.

Overly flexible similarity detection might easily become overwhelmed by false positives. But novel texts should not have highly similar prior art, while longer texts will quickly reduce the problem posed by false positives.

MRPC - Document Knowledge Graphs

The Microsoft Research Paraphrase Corpus (MRPC) contains pairs of sentences gleaned from news sources, with a judgement representing whether *"the two sentences to be close enough in meaning to be considered close paraphrases"* (Dolan & Brockett, 2005). Our working hypothesis is that sentence pairs may contain differing combinations of identical, similar and dissimilar elements. We treat the first sentence as a target whose novelty we wish to assess, while the second sentence is the closest identified prior art.

The MRPC is challenging as the similarity between sentence pairs is more nuanced than suggested by the binary categorization as either paraphrased (Para) or non-paraphrased (Orig). The paraphrase sentences contain significant amounts of differences while the non-paraphrased pairs also contain various differences. The MRPC includes a training set but this was not used to fine-tune our model.

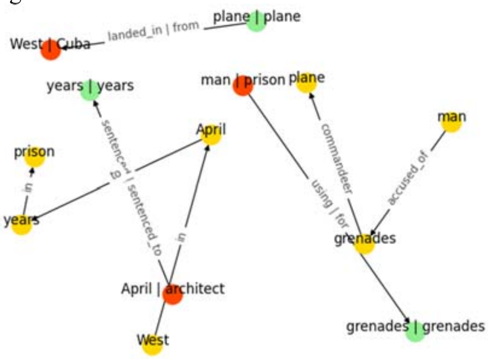


Figure 3: Red nodes map non-identical terms between sentences, revealing a possible instance of false novelty.

1458 pairs of graphs were extracted from 1641pairs of text, with failures often attributed to unsuccessful parsing of either sentence in a corpus pairs; eg *"The broader Standard & Poor's 500 Index <.SPX> was 0.46 points lower, or 0.05 percent, at 997.02."* Graphs contained an average of 3.8 edges (SD=3.4) ranging from 1 to 87 edges. There was a moderate difference between the sizes of the original and paraphrased graphs, with average sizes of 3.72 (SD=3.04) and 3.85 (SD=3.76) edges respectively. Paraphrased graphs were slightly larger and more diverse than the originals.

Figure 3 shows similarities between two sentence-graphs. The edge *(plane landed_in West)* was mapped with *(plane from Cuba)*. The items of greatest concern for plagiarism detection are the red nodes depicting paired non-identical concepts and the paired non-identical relations that are separated by "|". Of less concern are orange nodes showing unmapped concepts and green nodes indicate paired identical concepts. Identical paired edges are not repeated.

Quantitative Results for MRPC Sentence-Pairs

This analysis focuses on quantitative results, but each is accompanied by detailed lists of paired words or paired coreference chains, fostering deep expert or automated investigation of any discovered similarity. Table 1 *Para* indicates the similarity between paraphrased sentence-pairs, while *Orig* assesses Original (or non-paraphrased) sentence pairs. 95 pairs were identified as identical for the Para condition and just 30 for the Orig condition. Only 6 of 1482 sentence were identified by our system as having no detectable similarity. This indicates the prevalence of similarity between sentence pairs in this corpus, highlighting the challenge of distinguishing between them.

The average number of mapped edges for the Para condition was 2.65 and 2.18 for Orig. This moderate difference between the sentence types highlights that even Orig. sentence pairs contain much semantic overlap. The number of identical edges mapped in the paraphrase condition was 1.02 but only 0.58 for the Orig condition. Such overt similarity is not a source of concern for originality assurance.

The Para condition aligned 3.08 concept nodes on average, compared to just 2.80 for Orig. This quantifies the number of overt and latent similarities found. Over 1/3 of the graphs and approximately 50% of comparisons involved at least one node containing a coreference, showing the importance of intra-sentential coreferences.

Average Result	**Para**	**Orig**
Number of Identical Graphs	95.00	30.00
Avg. num. mapped edges	2.65	2.18
Avg. num. identical edges	1.02	0.58
Total S2v similarity	4.80	2.97
% total S2v similarity	0.53	0.44
Num. mapped concept nodes	3.08	2.80
Coreference Chain in mapping	0.38	0.24
% of target in LCC	0.58	0.66

Table 1: Comparison of MRPC sentence-pairs

We also estimated the semantic (sense2vec) similarity between mapped edges, each edge including two nouns and one verb. The average similarity for the Para condition was 1.60 but just 1.32 for Orig. from a maximum of 3. The Para condition accounted for 53% of the maximum possible similarity, while this was 44% for Orig. We identified the largest connected component (LCC) of the mapping. Surprisingly, the Orig sentences produced a stronger result, possibly indicating that further improvements are required.

Thus, Cre8blend identified a larger amount of stronger similarity between the paraphrased sentences than the non-paraphrased (Orig). We reiterate, these results are accompanied by detailed comparisons between the two sentences.

Qualitative Results for MRPC Sentence Pairs

We now illustrate some qualitative results from detecting latent similarity between potentially creative sentences and obfuscated versions that attempt to hide its true origins. Instances of questionable similarity between paraphrased sentences are presented. In the examples in this section, the aligned terms are generally located above one another allowing the non-literal similarity to be interpreted.

Synonym Replacement: Synonym replacement is a common strategy to feign novelty and avoid plagiarism detectors, but is detectable by our synta-semantic system.

*... by **two miles**, ... a seven-mile section ...*
*...by **three kilometres**, ... an 11-kilometre section ....*

Replacing multiple synonyms can also be detected.

*… to topple **Saddam** but to stabilize **Iraq**...*
*...to topple **Mr. Hussein** but to stabilize the **country**.*

Semantically Distant Term Replacement: Replacing multiple semantically distant words represents even greater challenge for plagiarism detection. Graph matching identified the following word-pairs: *replacement* ↔ *work; company* ↔ *officials.*

*The **company** didn't detail the costs of the **replacement** and repairs.*
*But company **officials** expect the costs of the replacement **work** to run into the millions of dollars.*

Unknown Term Introduction: Novel terms (like '5m' below) can also hide a documents' true origin but can be uncovered using context, such as aligning the following edges from 2 sentences: *(5m, over, violations), (million, settle, violations)*. We note also that this novel term was used in a somewhat dissimilar lexical context.

*PwC itself paid $**5m** last year ...*
*...PWC paid $5 **million** to settle alleged ...*

Questionable Similarity

We previously described three hallmarks of questionable similarity. Figure 4 depicts the results of applying one metric for questionable similarity to all sentence-pairs in the MRPC. We observe an exponential style distribution highlighting a small number of MRPC pairs displaying the three hallmarks of questionable similarity. While we cannot conclude these sentences are deliberate fakes, but we believe the authors of one of the following texts may be interested in the latent similarities identified by Cre8blend.

The highest questionable similarity score was for the following sentence pair, aligning 4 predicates and including 4 non-identical terms within that mapping.

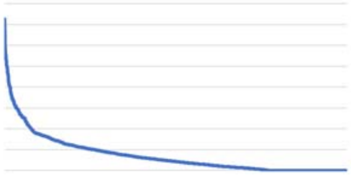


Figure 4: Few MSPR pairs have high questionable similarity

*Doctors who knowingly **violate** the **ban** could face up to two years in **prison**.*
*Under the measure, doctors who **perform** the **procedure** would be subject to two years in prison and unspecified **fines**.*

This pleasing result identified a large collection of parallels between the two texts, despite the small level of obvious similarity. The next highest result was for the following:

*Feith said **people** have **misconstrued** the **purpose** of the small intelligence review **team** he **assembled** in October*
*Feith said **critics** have **misrepresented** the **work** of the special intelligence **group** he **set** up in October*

This paired 4 edges from each graph, aligning 5 non-identical term-pairs between the two graphs. However, the next highest score can be considered a false positive arising from inaccurate identification of the predicate argument structure.

*They **were at** Raffles **Hospital** over the weekend **for** further **evaluation**.*
*They **underwent** more **tests** over the weekend, and are now warded **at** Raffles **Hospital**.*

Conclusions and Future Work

Our model successfully identified some instances of hidden similarity but requires further work with longer texts, as well as comparison to embedding and other approaches. A greater range of lexical information must also be extracted for the graphs. Refining our model may reduce instances of false positives, but its computational expense seems worthwhile only for valuable artefacts like publications, patents *etc* such as may arise from serious creativity. Examining suitable corpora may help identify typical similarity ranges for novelty assurance and for plagiarism detection.

Author Contributions and Acknowledgements

Diarmuid wrote the paper, Conor contributed broadly to similarity assessment, Shane contributed to the semantic evaluation and Gary contributed to graph matching. Other Maynooth contributors include Michael Griffith, Eoin Manning, Yalemisew Abgaz, Donny Hurley.

References

Abgaz, Y., O'Donoghue, D., Hurley, D., Chaudhry, E., & Zhang, J. (2017). Characteristics of Pro-c Analogies and Blends between Research Publications. *International Conference on Computational Creativity (ICCC).* Atlanta, GA, USA.

Aiyankovil, K. G., Monahan, R., & O'Donoghue, D. P. (2021). Upcycling Formal Specifications for Similar Implementations with Aris. *International Conference on Case Based Reasoning (ICCBR).*

Aiyankovil, K. G., O'Donoghue, D. P., & Monahan, R. (2021). Creating new Program Proofs by Combining Abductive and Deductive Reasoning. *International Conference on Computational Creativity (ICCC)*, (pp. 394-399). Meixco.

Boden, M. (1992). *The Creative Mind.* Abacus.

Carletti, V., Foggia, P., Saggese, A., & Vento, M. (2017). Introducing VF3: A new algorithm for sub-graph isomorphism. *International Workshop on Graph-Based Representations in Pattern Recognition*, (pp. 128-139).

Cordella, L. P., Foggia, P., Sansone, C., & Vento, M. (2004). A (sub) graph isomorphism algorithm for matching large graphs. *IEEE transactions on pattern analysis and machine intelligence, 26*(10), 1367-1372.

Dolan, B., & Brockett, C. (2005). Automatically constructing a corpus of sentential paraphrases. *Third International Workshop on Paraphrasing (IWP).*

Eppe, M., Maclean, E., Confalonieri, R., Kutz, O., Schorlemmer, M., Plaza, E., & Kühnberger, K.-U. (2018). A computational framework for conceptual blending. *Artificial Intelligence, 256*, 105-129.

Fang, Y., Wang, H., Zhao, L., Yu, F., & Wang, C. (2020). Dynamic knowledge graph based fake-review detection. *Applied Intelligence, 50*(12), 4281-4295.

Foltýnek, T., Dlabolová, D., Anohina-Naumeca, A., Razı, S., Kravjar, J., Kamzola, L. G.-D., . . . Weber-Wulff, D. (2020). Testing of support tools for plagiarism detection. *International Journal of Educational Technology in Higher Education, 17*(1), 1-31.

Gonçalves, J., Bembenek, A., Martins, P., & Cardoso, A. (2019). Going into Greater Depth in the Quest for Hidden Frames. *Intl. Conf. Computational Creativity ICCC*, (pp. 291-295).

Houbraken, M., Demeyer, S., Michoel, T., Audenaert, P., Colle, D., & Pickavet, M. (2014). The Index-based Subgraph Matching Algorithm with General Symmetries (ISMAGS): exploiting symmetry for faster subgraph enumeration. *PloS One, 9*(5).

Jordanous, A. (2012). A standardised procedure for evaluating creative systems. *Cognitive Computation, 4*(3), 246-279.

Markman, A. B., & Gentner, D. (1996). Commonalities and differences in similarity comparisons. *Memory & Cognition, 24*(2), 235-249.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546.*

Niklaus, C., Cetto, M., Freitas, A., & Handschuh, S. (2018). *A survey on open information extraction.* arXiv preprint arXiv:1806.05599.

O'Donoghue, D., Abgaz, Y., Hurley, D., Ronzano, F., & Saggion, H. (2015). Stimulating and Simulating Creativity with Dr Inventor. *International Conference on Computational Creativity (ICCC)*, (pp. 220-227). Utah, USA.

Pitu, M., Grijincu, D., Li, P., Saleem, A., Monahan, R., & O'Donghue, D. P. (2013). Arís: Analogical Reasoning for reuse of Implementation & Specification. *4th Artificial Intelligence for Formal Methods Workshop (AI4FM).* France.

Pollak, S., Podpecan, V., Kranjc, J., Borut Lesjak, & Lavrac, N. (2021). Scientific question generation: pattern-based and graph-based RoboCHAIR methods.

Prentice, F., & Kinden, C. (2018). Paraphrasing tools, language translation tools and plagiarism: an exploratory study. . *International Journal for Educational Integrity, 14(1), pp.1-16., 14*(1), 1-16.

Ramscar, M., & Yarlett, D. (2003). Semantic grounding in models of analogy: an environmental approach. *Cognitive Science, 27*(1), 41-71.

Rogerson, A., & McCarthy, G. (2017). Using Internet based paraphrasing tools: Original work, patchwriting or facilitated plagiarism? *International Journal for Educational Integrity, 13*(2).

Trask, A., Michalak, P., & Liu, J. (2015). sense2vec-a fast and accurate method for word sense disambiguation in neural word embeddings. *arXiv preprint arXiv:1511.06388.*

Vrbanec, T., & Meštrović, A. (2020). Corpus-based paraphrase detection experiments and review. *Information, 11*(5).

Weber-Wulff, D. (2019). Plagiarism detectors are a crutch, and a problem. *Nature, 567*(no. 7749), 435-436.

Zhou, Z. H. (2019). Fake news detection via NLP is vulnerable to adversarial attacks. arXiv preprint arXiv:1901.09657.

Word Embedding for Analogy Making

Mei Si

Department of Cognitive Science
Rensselaer Polytechnic Institute, Troy NY 12180, USA
sim@rpi.edu

Abstract

In recent years, natural language processing techniques have made impressive improvements in many tasks. However, their ability to make analogies is still minimal. This is partially due to the underlying representation of words and phrases, i.e., the word embedding is trained at the word sequence level and not at a concept relationship level. This work explores training a word embedding specifically for analogy making using knowledge graphs. The algorithm computes how analogous two concepts are based on the structural similarity of their adjacent concepts and relationships.

Introduction

Analogies describe comparative relationships between two sets of concepts. The Stanford Encyclopedia of Philosophy defines it as "An analogy is a comparison between two objects, or systems of objects, that highlights respects in which they are thought to be similar" (Bartha, 2019). With the recent release of large language models, such as GPT3 and BERT, natural language processing (NLP) algorithms can achieve almost human-level performance in some text generation tasks. For example, the AI Dungeon game is powered by GPT3 and can automatically generate dialogue and interactions with virtual characters as the user interact with the game. NLP algorithms have also achieved impressive performances in dialogue generation, question-answering, and even common sense reason tasks.

However, the current state-of-the-art NLP techniques still only have rudimental abilities in making analogies. A famous example of analogy-making came from Mikolov et al.'s work when the word2vec technique was invented for training word embedding (2013). Their work shows words that have similar meanings also have similar representations in the embedding space. Using vector operation, subtracting the embedding of the word Man from the embedding of King, and then adding the embedding of Women results in the embedding of Queen. I.e., the famous analogy example of:

King-Man+Women = Queen.

Another example of analogies formed based on word embedding is about locations. For example, the pair (Albuquerque, Albuquerque_Journal) is analogous to (Baltimore, Baltimore_Sun). While these analogies show that the trained word embedding is meaningful, they are not quite the same as typical analogies created by people. Further, more recent research on validating the analogies generated by word embedding found the system to be fragile and not always able to generate meaningful analogies. Even for the original example, “King-man+women” is actually closer to the embedding of King, rather than Queen (Nissim et al., 2019)!

The imperfection is not a surprise since the word2vec technique trains embedding using plain text, without exploring the relationships among concepts. In this work, we explore creating word embedding using algorithms inspired by cognitive theories of analogy, particularly the Structure-Mapping Theory (SMT) (Gentner, 1983; Gentner & Smith, 2012). SMT emphasizes the structural alignment of the relationships between two sets of concepts when forming analogies. We explore using structured content from knowledge graphs as input. The example outputs from our system show that the new embedding can create interesting and creative analogies among concepts.

Related Work

We review three types of related work: the cognitive theories about analogy-making, the knowledge graphs extracted from Wikidata, and a knowledge graph based analogy-making system.

Analogy-Making

How people form analogies has been studied extensively in cognitive science (Gentner, 1983; Kubose, Holyoak, and Hummel, 2002; Larkey and Love. 2003; Gentner and Smith, 2012). It is generally believed that analogy-making involves mapping concept groups with hierarchical structures from different domains.

The Structure-Mapping Theory (SMT) points out that analogical mapping is created by establishing a structural alignment of the relationships between two sets of concepts. The closer the structural match is, the more optimal the inferred analogy is. Surface features, i.e., properties of concepts that are not included in the hierarchical relationship structures, play little role in determining the analogy.

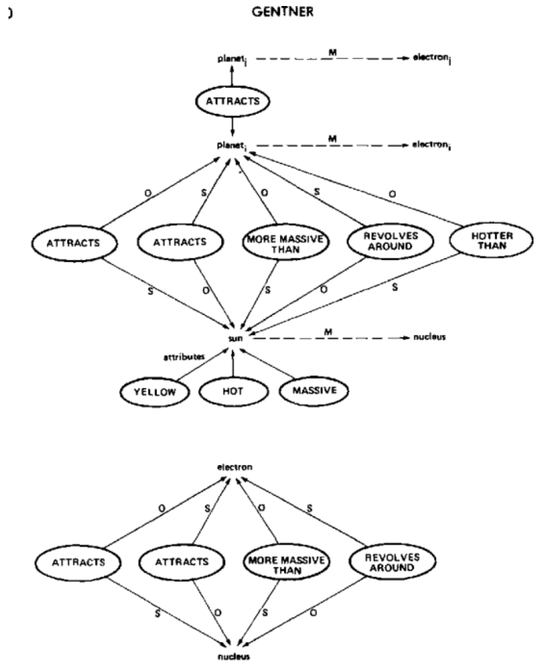


Figure 1: The analogy between the solar system and the Rutherford model (Figure taken from (Gentner, 1983).)

The Structure-Mapping Engine (SME) is a computational system that implements SMT (Falkenhainer, Forbus, and Gentner, 1989). A typical example produced by SME is the analogy between the Solar system and the Rutherford model, as shown in Figure 1. For producing this analogy, SME compares alternative ways of mapping the two groups of concepts to each other and determines that maximum structural mapping happens when the sun is mapped to the nucleus, and the planet is mapped to the electron. This mapping receives maximum support from the structural mapping of the relationships among these concepts. In the solar system, the sun and the planet have the "attracts" relationship in both ways, i.e., they both attract each other. The sun is also "more massive than" and "hotter than" the planet. The planet "revolves around" the sun. Similarly, in the Rutherford model, the electron and the nucleus have "attracts," "more massive than," and "revolves around" relationships. Furthermore, the "attracts" relationship results from both the sun and the planet having mass and gravity. The same relationship structure exists in the Rutherford model as well.

Structured Information in Knowledge Graphs

The input data -- the concepts and their relationships -- used by SME are manually designed as entities and predicates. To enable computer programs to generate analogies automatically, we also need to enable automation in generating input data. Knowledge graphs are composed of concepts connected by their relationships. They are structured data organized similarly to the manually curated data used by SME, and therefore provide a good basis for analogy generation algorithms.

Knowledge graphs cannot be directly used for computing structural mappings as in SME. Figure 2 provides an example knowledge graph crawled from Wikipedia. The main concepts from the solar system and Rutherford model analogy – sun, plant, electron, and nucleus were used as the seed nodes, and only concepts within two steps away from the seed nodes were included. The differences between Figures 1 and 2 are pretty obvious. The manually curated relationship structures only contain a limited set of entities. However, there are no natural boundaries for the groups of concepts when using knowledge graphs. This makes directly aligning two groups of concepts not feasible. Furthermore, the knowledge graph gathered from Wikipedia is less connected than the manually curated relationship structures. Typically, there is just one relationship between each pair of connected concepts. In contrast, as seen in Figure 1, there are often many relationships between a pair of concepts. In fact, these relationships are important supporting evidence when aligning the solar system and the Rutherford model.

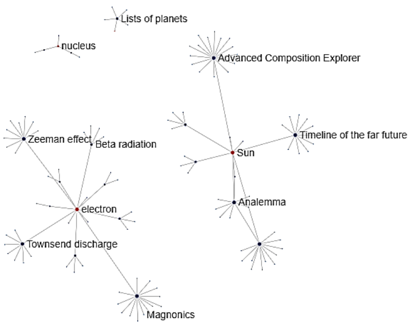


Figure 2: Sun and related concepts in Wikipedia.

Make Analogies using Knowledge Graph

This work is inspired by and based on (Si & Carlson, 2017), which uses information from DBpedia as the base for generating analogies. Si and Carlson's approach was inspired by the Structural Mapping Theory (SMT). The algorithm finds analogous relationship pairs, and the analogies are composed of a pair of mapping concepts and a set of supporting evidence, i.e., analogous relationship pairs.

An essential step in the algorithm is inferring pairs of analogous relationships. The algorithm computes how analogous two relationships are based on the topological similarity of their adjacent concepts and relationships. Si and Carlson compute four sets of relationship differences between the linked-from concept and the targeting concept:

1. Gain – what relationships are associated with the targeting concept but not the linked-from concept;
2. Loss – what relationships are associated with the linked-from concept but not the targeting concept;
3. Same – what relationships are associated with both the targeting concept and the linked-from concept;
4. Diff – the combination of the gain and the loss sets.

The differences among these sets are used to generate a unique index (embedding) for each relationship (Si & Carlson, 2017).

This relationship embedding serves as the basis for constructing analogies. If two concepts have many relationships that are analogous/similar to each other, the two concepts are regarded as being analogous. For example, Punk Rock is analogous to LPC (a programming language) because "the stylistic origin of Punk Rock is Garage Rock, Glam Rock, and Surf Music, just like LPC is influenced by Lisp, Perl, and C," and "Punk Rock is a music fusion genre of Celtic Punk, just like LPC influences Pike." Here, the analogy between Punk Rock and LPC is supported by mapping the "stylistic origin" of a music genre to the "influenced by" relationship among programming languages, and the "fusion genre" relationship among music genres to the "influence" relationship among programming languages. This approach mimics how structural mapping works in a weaker form.

Approach

This work explores an alternative approach for computing the embedding of relationships. Because the word2vec algorithm has been widely used for creating word embedding (Mikolov et al., 2013), we propose an algorithm that uses word2vec to compute the relationships embedding.

Our proposed approach contains three main steps, as illustrated in Figure 3. It first constructs a knowledge graph by crawling information from Wikidata. We use Wikidata instead of DBpedia to construct the knowledge graph.

For computing word embedding using word2vec, the words must appear in the input data many times. Only then the word2vec algorithm can learn their relationships with nearby words. Unfortunately, most concepts in Wikidata are unique, i.e., there is only one entry for each concept. Therefore, the word2vec algorithm cannot be directly applied. On the other hand, the relationships in Wikidata are rarely unique. E.g., "Give Name" is a popular relationship that connects many pairs of concepts. Therefore, in the second step, we construct a reversed knowledge graph where the relationships are nodes and the concepts are edges, as shown in Figure 4. And finally, we compute the embedding for the relationships using this reversed knowledge graph.

Construct Knowledge Graph

For getting information from Wikidata, we used a web crawler, which stores concepts and their relationships in a network structure. For creating the knowledge graph we used in this work, we used 18 seed words, and did a breadth-first search around each of them until at least 1000 nodes had been reached. Then we merged all the data collected. The resulting knowledge graph contains 219691 entities and 1540 unique relationships.

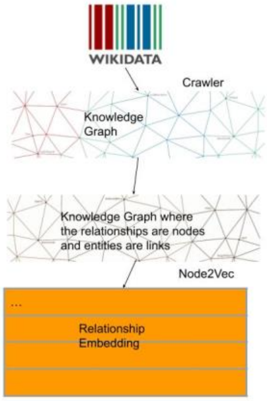

WIKIDATA
Crawler
Knowledge Graph
Knowledge Graph where the relationships are nodes and entities are links
Node2Vec
...
Relationship Embedding

Figure 3: Workflow.

For computing the embedding for the relationship, we built a reversed graph where the relationships are nodes, and the entities are links. For example, in Wikidata, "member of political party" is the relationship between "Armen Sarkissian" and "independent politician." In this reversed graph, relationships such as "member of political party" and "given name" become nodes, and the entities become edges. We then apply the node2vec algorithm on this graph to obtain the embedding for the relationships (Grover & Leskovec, 2016).

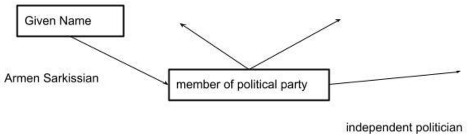


Figure 4: Reversed Knowledge Graph.

Node2vec

Node2vec is an embedding algorithm developed by Grover & Leskovec (2016). This algorithm can convert nodes in a graph into numerical representations, i.e., embedding. Node2vec works in two steps. The first step uses a second-order random walk on the graph to generate transaction samples. These samples are equivalent to the text input to word2vec, and the second step uses word2vec to compute the embedding. Take Figure 4, for example; the random walk algorithm would visit each note multiple times, and randomly follow a link to move to the next node each time. After sampling, the graph is essentially converted to a list of linear transactions, each of them contains a list of nodes, e.g. [given name, member of political party …]. These linear transactions become the corpus for word2vec.

Example Output

Like regular word embedding, the relationship embedding computed in this work allows us to calculate the distance between two relationships and find the most similar relationships. We also implemented the algorithm from (Si & Carlson, 2017) and compared these two embeddings.

Both embeddings are not perfect but can provide some insightful results. Moreover, their results read more like figurative language than a simple word association. For example, Tables 1 and 2 list the top 10 closest relationships to two relationships we used for testing. The closest ones are on the top.

Table 1: Results for "member of political party".

(Si & Carlson, 2017)	Node2Vec
Work location	Family name
Place of birth	Military rank
Place of death	Position held
Language used	Military branch
Official Language	Sibling
Residence	Spouse
Place of burial	Moth
Educated at	Native language
Parent astronomical body	Educated at
Country of citizenship	Sex or gender

Table 2: Results for "architectural style".

(Si & Carlson, 2017)	Node2Vec
Origin of the watercourse	Architect
Director/Manager	Heritage designation located on street
Material used	Drainage basin
Occupant	Material used
Lyrics by	Legal form
Anthem	Located on terrain feature
Legislative body	Located in time zone mouth of the watercourse
Legal form	Contain settlement
Currency	
Industry	

In Table 1, both embeddings suggest "Educated at" could be an analogy to "member of political party." And in Table 2, both suggest "Legal form" could be an analogy to "architectural style." We think these suggestions are pretty creative.

Note that compared to WikiData itself, our crawled dataset is tiny and sparse. Therefore, these suggested relationships are not necessarily the best analogies from people's points of view. Nevertheless, most proposed relationships convey meaning more or less similar to the source concept.

Discussion and Future Work

We aim to create analogies where the relationship mapping itself is analogous. Though the process of computing how analogous two relationships are to each other leverages the idea of computing structural similarity, we suspect the results presented here are different from results produced by SME or other systems that infer analogies solely based on structural similarities. Using SME, the symbolic meanings of the relationships are discarded, and only the structural alignment between the two groups of concepts is considered. Two relationships both named involving do not make them more analogous to each other than two relationships with different names. In our results, the meanings of the relationships are undoubtedly important. We plan to explore this phenomenon and exam further to what degree the embedding we computed is independent of the relationships' symbolic meanings in the future.

The current work finds analogous relationships, but does not use them to find analogous concepts yet. We will explore this direction in future work. We are also interested in computing the relationship embedding using a larger knowledge graph and seeing whether that improves the results.

Author Contributions

M Si ideated and wrote the paper alone.

Acknowledgements

The author wants to thank the anonymous reviewers for their thoughtful comments and suggestions.

References

Bartha, Paul: Analogy and analogical reasoning. In: The Stanford Encyclopedia of Philosophy. Spring 2019 ed. Edward N. Zalta (ed.), forthcoming URL = https://plato.stanford.edu/archives/spr2019/entries/reasoning-analogy/

Falkenhainer, B., Forbus, K., Gentner, D.: The structure-mapping engine: Algorithm and examples. Artificial Intelligence, 41, 1–63. (1989).

Forbus, K., Oblinger, D.: Making SME greedy and pragmatic. In: Proceedings of the 12th Annual Conference of the Cognitive Science Society, 61–68. (1990)

Gentner, D., & Markman, A. B.: Structure mapping in analogy and similarity. American Psychologist, 52, 45-56. (1997).

Gentner, D., Smith, L.: Analogical reasoning. In V. Ramachandran (Ed.), Encyclopedia of human behavior. 2nd ed. pp. 130–136. Elsevier; Oxford, UK. (2012).

Gentner, D.: Structure-mapping: A theoretical framework for analogy. Cognitive Science, 7 (2), 155–170. (1983).

Grover, A., & Leskovec, J. (2016, August). node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 855-864).

Kubose, T. T, Holyoak, K. J, Hummel, J. E.: The role of textual coherence in incremental analogical mapping. Journal of memory and language, 47(3), 407-435. (2002).

Larkey, L. B., Love, B. C. CAB: Connectionist analogy builder. Cognitive Science, 27(5), 781-794. (2003).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).

Nissim, M., van Noord, R., & van der Goot, R. (2020). Fair is better than sensational: Man is to doctor as woman is to doctor. Computational Linguistics, 46(2), 487-497.

Si, M., Carlson, C.: A Data-Driven Approach for Making Analogies. In: Proc. Cognitive Science Society Conference, pp. 3155-3160. (2017).

Mechanising Conceptual Spaces using Variational Autoencoders

Max Peeperkorn[1], Rob Saunders[2], Oliver Bown[3], Anna Jordanous[1]
[1] School of Computing, University of Kent, Canterbury, United Kingdom
[2] Leiden Institute for Advanced Computer Science, Leiden University, Leiden, Netherlands
[3] Faculty of Art and Design, University of New South Wales, Sydney, Australia
mp770@kent.ac.uk

Abstract

In this pilot study, we explore the Variational Autoencoder as a computational model for conceptual spaces in a social interaction context. Conceptually, the Variational Autoencoder is a natural fit for this purpose. We apply this idea in an agent-based social creativity simulation to explore and understand the effects of social interactions on adapting conceptual spaces. We demonstrate a simple simulation setup and run experiments with a focus on establishing a baseline. While ongoing work needs to identify if adaption was appropriate, the results so far suggest that the Variational Autoencoder appears to adapt to new artefacts and has potential for modelling conceptual spaces.

Introduction

In society, humans share their ideas and exchange artefacts. We draw inspiration from these interactions, and this sparks our imagination to produce new ones (Vygotsky 2004). Every individual has a unique perspective, a style of thought, embedded in a conceptual space (Boden 2004). While ideas and artefacts can be attributed to individuals, they are shaped by others, leading to a distributed emergence of creativity.

In this paper, we explore the use of the Variational Autoencoder (VAE) (Kingma and Welling 2014) as a computational model for the conceptual space in an artificial social context. This is an initial study investigating how to embed and maintain VAEs in an agent-based Computational Social Creativity (Saunders and Bown 2015) simulation.

Background

Conceptual Spaces... There are two views on conceptual spaces: a creativity view (Boden 2004) and a general cognitive view (Gärdenfors 2004). Gärdenfors proposed conceptual spaces as a geometric mental structure to organise thought, with the aim to bridge the symbolic and the subsymbolic. It allows finding similarities between symbols that cannot be derived from the symbolic level alone. According to this theory, concepts are convex regions in the conceptual space, and the axes represent properties. Boden's view of the conceptual space is well-known and a central part of her creativity framework concerning the three modes of creativity. This definition is abstract and less defined, simply the set of artefacts that follow the rules of a given domain. While useful to reason about creativity, Boden's abstract definition is unsuited for computational purposes. However, in this paper, we are less concerned with the formal definition and use both views to inform our choices in the simulation. We use Boden's view to examine the creative act and use Gärdenfors' view to inform traversing the conceptual space.

...and Variational Autoencoders Due to its probabilistic nature, and compression and generative capabilities, we explore the idea that VAE is conceptually a natural fit for approximating conceptual spaces. The VAE is a deep generative model that learns fuzzy relations in the data and maps this onto smooth latent spaces—which is reminiscent of Gärdenfors' geometric conceptual space. The latent space can be queried to find similar artefacts and sampled to generate new artefacts. This makes it particularly interesting to use as a way for agents to perceive, interpret, and produce artefacts. Based on its characteristics, we assume that the VAE is a reasonable abstraction for the formation of concepts and properties.

Simulation

Like other simulations of social creativity (Saunders 2012), the DIFI model (Feldman, Csikszentmihalyi, and Gardner 1994) provides the conceptual model for the simulation presented here. To explore how to embed and maintain the VAEs in a simulation, we use them in two ways: as the conceptual space for each agent and as a recommender system for the whole domain. Next, we discuss the data representation, VAE architecture, each component of the DIFI model, and further discuss the details of the utility of VAE in the simulation.

Data Representation

For use in the simulation, the VAEs require pre-training that can be likened to providing basic education for each agent. Initially, we used a generated dataset in a simplified musical domain of short melodies of 16 timesteps of 12 pitches (chromatic scale) (Peeperkorn, Bown, and Saunders 2020). Further work proved this dataset to be problematic and led to heavy overfitting when pre-training the VAEs. To mitigate this, we generated a dataset using Hidden Markov Mod-

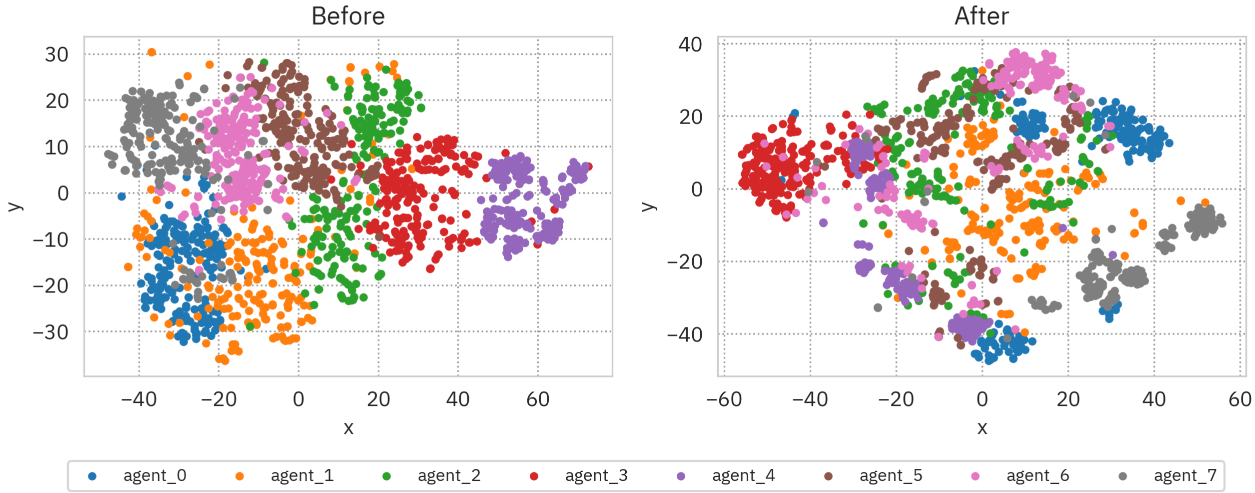


Figure 1: Sampling from Agent VAEs before and after the simulation and compared in the Domain VAE projected using T-SNE.

els fitted to real music data.[1] Subsequently, we generated a combined dataset of 400k samples of 16 timesteps and 88 pitches. We considered other datasets, such as images of typefaces, but the benefit of using categorical data is that it allows for exact reconstructions.

Recurrent VAE architecture

We used a simple recurrent VAE (Fabius and Van Amersfoort 2014) using Long Short-Term Memory (LSTM) layers. A big issue with recurrent VAEs is posterior collapse which occurs when the network learns to ignore the latent space. The Kullback-Leibler (KL) term is annealed in the early stages of the training (Bowman et al. 2015) to mitigate this issue allowing the VAE to extract informative features before the full penalty smooths the latent encodings. The final VAE network has a 32-dimensional latent space. The encoder and decoder consist of two hidden LSTM layers with 128 nodes. For initial training, we used a batch size of 512 and KL-annealing over the first 200 epochs.

DIFI model Setup

Domain The domain is explained as a cultural repository of knowledge (Csikszentmihalyi 2014). In this work, there is no single repository for agents to access. Instead, the domain is distributed amongst the agents' conceptual spaces, each with a personal subset of embedded knowledge. This does not allow artefact comparison on the individual level, and therefore, we introduce a static and pre-trained Domain VAE. It operates as an archimedean point that enables the analysis of the distributed domain. Additionally, the Domain VAE is used to split the dataset into different slices for each agent using a 2D PCA projection of the latent encodings.

Individual Each agent in the simulation has a personal VAE, each trained on a different slice. In contrast to the Domain VAE, the individual agent uses the VAE to learn from and generate new artefacts. Generating is done by randomly sampling from a gaussian distribution, and decoding the latent vector to produce the artefact. We assume that the standard deviation can be used as a proxy for novelty preference. A narrow distribution produces less varied artefacts, and conversely, a wide distribution produces high variation.

Field The field acts as a gatekeeper for what artworks are selected for circulation, according to the ideology of society (Csikszentmihalyi 2014). Different ideologies use different selection criteria, and subsequently, influence the social interactions taking place in the domain. The field acts according to an ideology, a social policy, for selecting artefacts for the next round in the simulation. In the current setup, we use a neutral policy, i.e. that every artefact has an equal chance of being "put on display" in the field. The Recommender System (Domain VAE) informs the field of its choices. As such, the field fulfils two roles in the model: the matchmaker and the gatekeeper. The matchmaker takes the newly produced artefacts and determines the agent's position to find neighbours who share their artefacts. Subsequently, each agent has a different pool from which the gatekeeper will select for the next round.

Interaction After initialising the VAEs, the simulation iterates through three stages. The first stage is associated with the individual, where the newly observed artefacts are used to fine-tune the agents' latent space for a given learning budget to extract new features and then produce several new artefacts sampled according to the novelty preference. The second stage is where the field receives the position of each agent, queried from the Recommender System using the mean of the newly produced artefacts. In the third stage, the positions are used to determine the agents' nearest neighbours. The neighbour shares their artefacts, which form a pool of artefacts. Subsequently, the field selects artefacts from this pool for the next round according to its ideology.

[1] The data is gathered from the Humdrum database (https://kern.humdrum.org), selecting the 8 genres with the most samples.

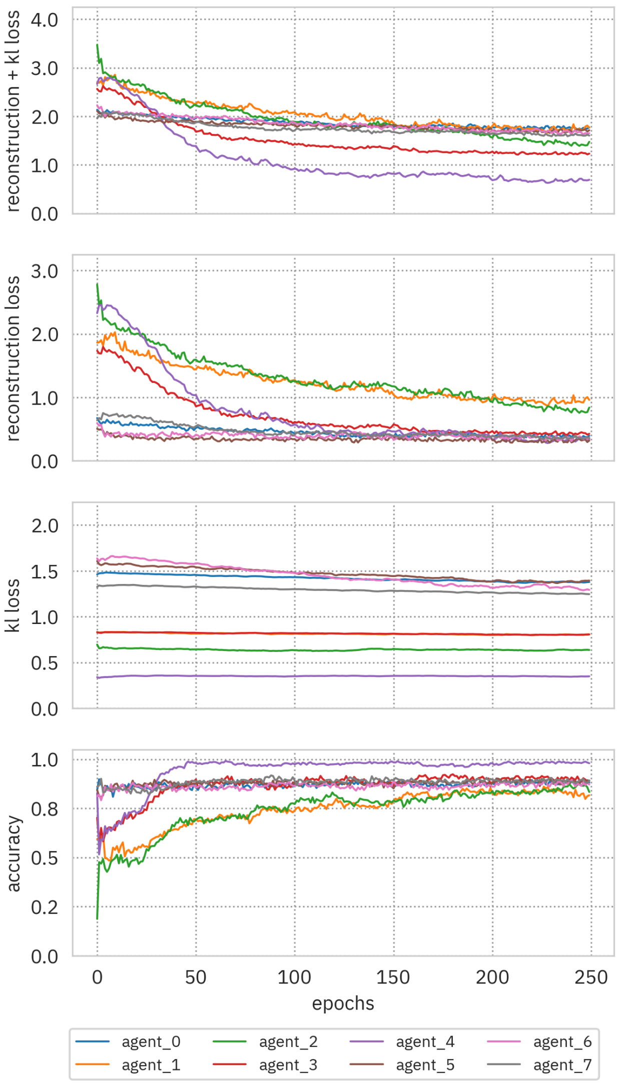


Figure 2: Agent VAE performances evaluating artefacts over a sliding window of 25 epochs.

Results

The simulation experiments use the following settings: 250 epochs with 8 agents, the neutral ideology, and novelty preference set to 0.25. Each round, the field selects 128 artefacts, individuals produce 4 new artefacts, and 1 neighbour shares their artefacts. Each agent has a 5-epoch budget for fine-tuning using a learning rate of 10^{-4}.

The VAEs are trained on the respective datasets using a 70/30 train/validation split. Table 1 shows that Domain VAE performs very well. The agents show clear clusters after the initialisation (Fig. 1). However, the agent VAE pre-training show very mixed results and some perform well (>80% accuracy), while others do not (<30% accuracy).

Post-simulation sampling of the agent VAEs suggests that they mingled as expected (Fig. 1). However, there are a few very dense clusters, which could signify that the latent space is collapsing.

Table 1: Pre-training results show loss and accuracy after 2000 epochs. The Agents VAE shows the mean results for 8 agents.

	Loss		**Accuracy**	
	Train	Val	Train	Val
Domain VAE	2.028	2.034	.937	.934
Agents VAEs	2.593	2.894	.559	.497

The results in Fig. 2 on the other hand, appear to indicate that agents adapt well, within a 25-epoch sliding window, to the artefacts selected each round as accuracy goes up and reconstruction loss goes down. It is somewhat surprising given the agent initialisation results (Table 1). While this is desirable, it might also indicate overfitting. The KL loss is level, suggesting latent space stability, but an issue is that, for some agents, it is already very low after pre-training.

Discussion

The results suggest the conceptual spaces drift stably, which, in turn, suggests that the VAEs adapt. However, it does not inform to what extent they adapted and if it is appropriate according to the social dynamics and interactions. With the current setup, it is very difficult to observe exact agent behaviours. Crucial for future work is to further investigate VAE performance during the simulation and rule out the previously mentioned issues, such as overfitting or posterior collapse. Even though the VAEs appear operable, the performance still causes some concern. It could be due to the datasets, but it might also be that the domain requires a more sophisticated VAE architecture, such as the Hierarchical decoder (Roberts et al. 2018).

This paper focuses on getting the VAE to work and less on the social dynamics. It does provide opportunities for examining different novelty preferences or ideologies, for example, progressive (seeking novelty) and conservative (seeking familiarity). These research directions are interesting to explore, but they depend on the ability to look inside the simulation and inspect the VAE behaviour. The main challenge remains: to develop the tools leveraging latent traversals to increase understanding of how the VAE behaves throughout the simulation. This is necessary to see if social dynamics and interactions explain agent VAE divergences. But this work establishes an initial baseline for future work.

Conclusion

The work presented here is an initial study into mechanising conceptual spaces using VAEs. The results suggest the potential for the VAE as a computational model for conceptual spaces. We stress that additional sophisticated analysis is necessary to further examine the VAE behaviours. However, it shows the potential of VAEs for modelling ill-defined domains without predetermined rules, which is so often the case with creative domains.

Author Contributions

MP designed the simulation and experiments with RS. MP implemented the simulation and experiments. MP and AJ contributed to improving the pre-training dataset setup. RS, OB, and AJ contributed valuable ideas, insights and supervisory feedback. MP wrote the paper. MP and AJ edited the paper.

Acknowledgements

We would like to thank Marek Grześ, Piotr Sawicki, and Dan Brown for their insightful suggestions and comments. This work is a continuation of the MP's master thesis supervised by RS and OB for the Media Technology MSc programme at Leiden University, NL. MP is supported by the University of Kent GTA Studentship Award, Prins Bernhard Cultuurfonds, Hendrik Mullerfonds, and Vreedefonds.

References

Boden, M. 2004. *The Creative Mind: Myths and Mechanisms*. London: Routledge.

Bowman, S. R.; Vilnis, L.; Vinyals, O.; Dai, A. M.; Józefowicz, R.; and Bengio, S. 2015. Generating sentences from a continuous space. *CoRR* abs/1511.06349.

Csikszentmihalyi, M. 2014. *Society, Culture, and Person: A Systems View of Creativity*. Dordrecht: Springer Netherlands. 47–61.

Fabius, O., and Van Amersfoort, J. R. 2014. Variational recurrent auto-encoders. *arXiv preprint arXiv:1412.6581.*

Feldman, D. H.; Csikszentmihalyi, M.; and Gardner, H. 1994. *Changing the world: A framework for the study of creativity*. Westport: Praeger Publishers.

Gärdenfors, P. 2004. *Conceptual spaces: The geometry of thought*. MIT press.

Kingma, D. P., and Welling, M. 2014. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR) 2014*, volume abs/1312.6114.

Peeperkorn, M.; Bown, O.; and Saunders, R. 2020. The maintenance of conceptual spaces through social interactions. In *Proceedings of BNAIC/BENELEARN 2020.* Master Thesis Abstract.

Roberts, A.; Engel, J.; Raffel, C.; Hawthorne, C.; and Eck, D. 2018. A hierarchical latent vector model for learning long-term structure in music. In *International conference on machine learning*, 4364–4373. PMLR.

Saunders, R., and Bown, O. 2015. Computational social creativity. *Artificial Life* 21(3):366–378.

Saunders, R. 2012. Towards autonomous creative systems: A computational approach. *Cognitive Computation* 4(3):216–225.

Vygotsky, L. S. 2004. Imagination and creativity in childhood. *Journal of Russian & East European Psychology* 42(1):7–97.

Competitive Language Games as Creative Tasks with Well-Defined Goals

Brad Spendlove and Dan Ventura
Computer Science Department
Brigham Young University
Provo, UT 84602 USA
brad.spendlove@byu.edu, ventura@cs.byu.edu

Abstract

Creative computer systems grapple with challenging tasks that exist within effectively endless combinatorial spaces. Further complicating these already difficult tasks is the fact that the goal of high-quality creative output is itself nebulous. A creative domain with concrete goals would therefore be a fruitful domain for studying computational creativity. We propose that competitive language games are just such a domain—they require creativity but also feature concrete win and loss states. We present an analysis of creative agents that play one such game: Codenames, a 2016 board game of communicating hidden information via single-word clues. Our model-agnostic framework allows us to compare agents that utilize different language models. We present our findings and discuss how future computational creativity research can continue to explore competitive language games.

Introduction

AI agents pursue a goal within an environment. Creative computational (CC) systems are AI agents that seek to generate or identify high-quality creative artifacts within the environment of an effectively endless combinatorial space. A plethora of potential output artifacts exists within that space, each with varying levels of quality. CC systems, therefore, often contend with the unique challenge of seeking a goal that is not well defined.

The space of all possible artifacts for any given human creative domain is so large that defining a goal for a creative agent can be as difficult as building the agent that pursues that goal. Because human creativity is extremely complex, and its mechanisms are only partially understood, CC systems' goals must necessarily be abstractions. The degree to which those abstractions represent the goals of real-world creativity corresponds to the maximum creative potential of systems that use them.

Thus, seeking or developing better defined creative goals is a fruitful avenue for computational creativity research. Enter board games. Concomitant with the boom in modern board gaming (Jolin 2016) is the rise of new social, language-based games in which participants use their creativity to come up with clues, guesses, and deceptions.

Classic guessing games such as Guess Who and newer games like Mysterium (Nevskiy and Sidorenko 2015) require players to reason and make verbal guesses about images. Hidden role games like Werewolf and Spyfall (Ushan 2017) are freer form and involve players talking with one another to deduce others' hidden roles while keeping their own a secret. These games all involve reasoning, creativity, and language skills but critically also include clear objectives and win/lose states. Playing these games is a creative task with the well-defined goal of winning the game. We propose that they are therefore ideal candidates for computational creativity research.

In this paper, we present and analyze a creative system that plays Codenames (Chvátil 2015), winner of the prestigious Spiel des Jahres in 2016.[1] Codenames is a word-based guessing game in which two teams play on a shared grid of 25 word cards drawn randomly from a large deck. One player from each team serves as a "spymaster" who must give their teammates one-word clues corresponding to certain words on the board that are assigned to each team, secret to all except the spymasters. Clues are phrased as a single word and a number, indicating how many cards the clue is intended to relate to.

The teammates then discuss the clue and select word cards on the grid to guess one at a time until they either guess incorrectly or pass. A correct guess identifies one of the team's assigned words. An incorrect guess accidentally identifies one of the opposing team's words, a neutral word belonging to neither team, or an "assassin" word that results in instant game loss. Teams take turns giving clues and guessing until one team wins by identifying all of their assigned cards (perhaps with inadvertent assistance from their opponents) or the opposing team guesses the assassin.

Figure 1 shows an example of a Codenames board of 25 word cards. Previously guessed words are covered with colored tiles corresponding to their hidden roles: blue and red for the opposing teams, grey for neutral, and black (out of frame) for the assassin.

The spymaster's role is to come up with one-word clues that elegantly identify multiple correct words while excluding incorrect words. Importantly, the spymaster's clues are not restricted in any way other than by simple rules about not using words on the cards or acronyms, etc. This task requires knowledge of what each word means and how they

[1] https://www.spiel-des-jahres.de/spiele/codenames/

Figure 1: An example Codenames board, showing covered and uncovered word cards.
Credit: Skip McIlvaine, boardgamegeek.com, CC0 license.

relate to one another. The role of spymaster is easy to attribute creativity to—clues must often navigate tricky positive and negative relationships, and human players can recognize particularly clever or helpful clues.

The spymaster's teammates who guess based on the clues have a less open-ended task, as it is restricted to linking the clue word to one word card at a time. This role requires relatively less creativity, but it is still a non-trivial language task for humans and computer systems.

In order to complete either of these tasks, an agent must have an understanding of how words both positively and negatively associate with one another. Using that knowledge, the spymaster searches for a clue word that their guessers will most likely associate with a chosen subset of the team's word cards while avoiding associations with incorrect word cards. The guesser uses a similar language faculty to guess word cards that most closely relate to the clue.

In this paper, we present a framework for a system that completes those tasks to play Codenames in both the spymaster and guesser roles. We explore how different models perform at these tasks in competition against one another, with the goal of demonstrating how competitive language games can serve as useful test beds for creative systems.

Creativity in Codenames

The spymaster's task involves elements of puzzle and problem solving, and we may view the potential for creativity in solving the task through that lens. The skills required of a good spymaster player are related to problem framing (Guilford 1956; Dorst 2011) and re-representation (Ohlsson 1992; Veale 2006), both of which are well-known to facilitate creativity.

It is important to realize that even though the output produced by the spymaster is a single clue word (and an associated number), the artifact which the spymaster is creating is not a word. Rather, it is something like a multi-word relationship graph (including both positive and negative connections). The spymaster uses skills such as those mentioned above, as well as, of course, their knowledge of language to create this graph structure, which, in a more traditional creative setting, would constitute the output artifact. To make this a game, the artifact is instead obfuscated, with only the clue word and number giving hints about its structure. The guessing players' job is, essentially, to re-create this relationship graph from the clues and use it to identify words assigned to their team.

Serendipitously, it is this gamification of creativity that affords us a well-defined, if indirect, measure of creativity in the form of game outcomes. While in many traditional creative settings, artifact value is often very difficult to measure, the appeal of Codenames—and other competitive language games—is that the value metric is (at least) strongly correlated with the win/loss outcome. This serves as a powerful proxy for evaluating the creativity of the system itself, or at least one critical element of it.

Creative domains are characterized by their extremely large combinatorial spaces, which are a prerequisite for novelty. The word relationship graphs that are key to playing Codenames are simpler than other types of artifacts such as literature or visual art, but they are nevertheless complex enough to be considered in the same class as metaphors or short jokes and witticisms—domains which make a strong case that creativity can be manifest where artifacts take the form of a small number of words connected in a clever or surprising way.

Related Work

We use language models to provide the word association faculties that our Codenames player agents need. In particular, we use word2vec and GPT-2. We make use of word2vec's word embeddings and both GPT-2's word embeddings and text generation capabilities.

Word embeddings are a way to represent words as vectors such that vectors that are close to each other occur in similar contexts in text. The distributional hypothesis posits that words that occur in similar contexts have similar semantic meanings (Harris 1954). This hypothesis is the basis for distributional semantics (Sahlgren 2008), a theory that forms the basis for word embedding models. A model built on this theory is a natural fit for use in playing Codenames because words that appear in similar contexts are likely to be associated with one another in a way that will help players guess related words.

A word embedding model is trained on a corpus of text to encode the relative contexts of the words in the corpus into a vector space. Word2vec (Mikolov et al. 2013a) is a neural network model that learns word associations in this manner. The word2vec model we use implements a skip-gram and negative sampling unsupervised learning model, which learns to predict the context for a given word in the corpus. It is trained to build an embedding that minimizes the distance between a word and its context while maximizing the distance between a word and a hallucinated (random) context. The weights that are trained with this method are treated as the vector space into which words are embedded.

GPT-2 (Radford et al. 2019) is a powerful, large language model (LLM) that implements a transformer architecture (Vaswani et al. 2017). This unsupervised training method uses attention mechanisms to focus learning on small but important parts of the training corpus. GPT-2 has been demonstrated to perform well at a variety of tasks such as summarization, translation, question answering, and text generation. It is notable, however, that the model was not trained on any of those tasks explicitly. Its attention-based language model learns implicitly to complete such tasks via training to predict text from a prompt.

GPT-2 can be used (with varying degrees of success) as a general-purpose model by providing it a text prompt that describes a task to be completed. The model generates output that it predicts to be a likely continuation of the prompt. This output is highly dependent on the prompt, giving rise to a new and still-developing discipline known as prompt engineering (Liu et al. 2021). A common form for LLM prompts is listing a handful of complete examples of the task to be solved and then providing an incomplete task for the model to complete.

Prompt engineering is now an integral part of natural language processing using LLMs. Recent projects like PromptSource (Bach et al. 2022) facilitate the sharing of prompts for various tasks, allowing the research community to build upon past successes and find more useful prompts.

We built our word2vec agents with the Gensim implementation (Řehůřek and Sojka 2010), using "pre-trained vectors trained on part of the Google News dataset (about 100 billion words)" (Mikolov et al. 2013b). Our GPT-2 agents used HuggingFace GPT-2 Small (124M parameters) (Wolf et al. 2020).

Methodology

To experiment with using Codenames as a test bed for creative language systems, we built an agent-agnostic game playing framework and implemented language model-agnostic AI player agents for both the spymaster and guesser roles. As this work is focused on language-based creativity, we use the same rudimentary decision-making process for all the agents we experimented with. There are undoubtedly many possible improvements to their strategies, but they are reasonable for the purposes of this research. By keeping the agents' strategies static, we are better able to isolate the effect of using different language models.

All of the code described in this section can be found in a public GitHub repository. [2]

Defining the Codenames Task

The Codenames game board is a set G of 25 word cards drawn from a deck of size 400 (for play, the cards are arranged in a 5x5 grid). G is partitioned into four subsets: T, an unknown set of target words, P, an unknown set of opponent words, N, an unknown set of neutral words and $A = \{a\}$, an unknown singleton set that contains an assassin word. $G = T \cup N \cup P \cup A$ represents an instance of the game with $|G| = 25, |T| = 9, |P| = 8, |N| = 7, |A| = 1$.[3]

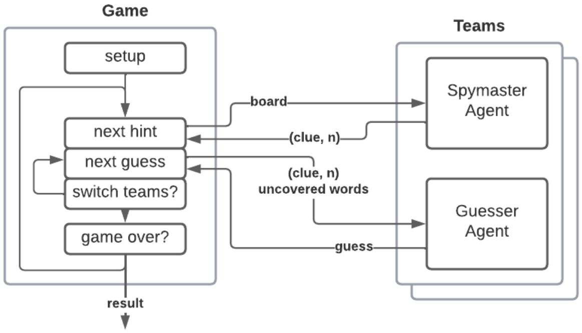


Figure 2: The high-level information flow of our Codenames framework.

While the set of cards G is public knowledge, its partitioning is not; this information is initially only available to the two teams' spymasters, who have access to a secret key.

Let U be the set of cards whose partition membership is currently unknown. Initially, $U = G$. The object of the game is for the spymaster to help their teammates discover which words are in T before the opponent discovers which words are in P and without discovering the identity of word a. Teams alternate playing in rounds. Whichever team plays first will have nine words to guess, while the other will have eight (note that for convenience and without loss of generality, we assume $|T| = 9$).

Play proceeds in the following manner. The active team's spymaster generates a clue $c = (w, k)$ consisting of a clue word $w \in W \setminus G$ and a number $0 < k \leq |T|$, where W is the set of all English words[4] and $k = |I|$ where $I \subseteq T$ is a secret set of words to which the spymaster intends the clue to correspond. I itself changes every round, depending on the spymaster's strategy, and is not recorded in the game.

Given a clue word w, the guesser may then make a maximum of $k + 1$ guesses. The guesser's task is to guess a word $v \in U$ whose partition membership is then revealed (by covering it with one of four tile types), removing it from U (by removing it from its secret partition). If $v \in T$, the team guessed correctly and may pass or guess again as long as they have not exceeded $k + 1$ guesses for the current round. The round is over if the guesser has used all of their guesses, if they pass, if $T = \emptyset$, or if $v \notin T$. If $v = a$ or $P = \emptyset$ (meaning they guessed the assassin word or their opponents' final word), the team loses the game. If $T = \emptyset$, they win the game. Otherwise, play passes to the other team. This process of playing rounds repeats until one team wins.

Codenames Framework

We built a framework for playing teams of Codenames agents against one another. Figure 2 shows a diagram of our framework's architecture. The Game module randomly determines which team will play first, sets up the board and secret key, and begins the gameplay loop. In

[2] https://github.com/gbspend/codenames

[3] This admits $\binom{400}{25} = 3.374984143967 \times 10^{39}$ unique draws, each of which admit $\binom{25}{9}\binom{16}{8}\binom{8}{7} = 2042975 \times 12870 \times 8 = 210,344,706,000$ possible games, for a total of $7.099100475174 \times 10^{50}$ unique games.

[4] We assume W excludes acronyms and proper nouns.

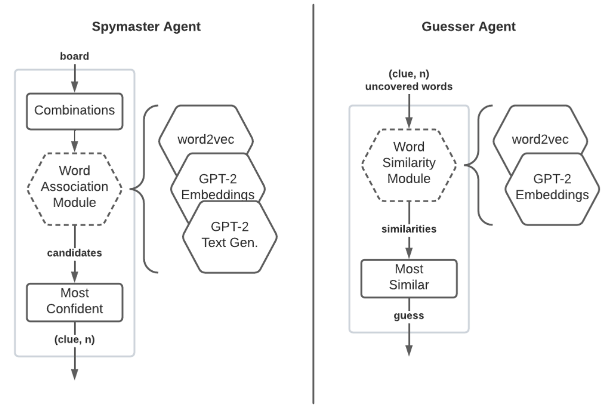


Figure 3: Information flow in our two types of Codenames agents. Note that the language task modules are pluggable.

each round, the Game module passes the current board state $b = (T, P, N, A)$ to the active team's spymaster. The spymaster agent returns a clue $c = (w, k)$. That is, the spymaster agent implements a function $\sigma : \mathcal{B} \to \mathcal{C}$, where $\mathcal{B} = 2^T \times 2^P \times 2^N \times 2^A$ is the set of possible game board states[5] and $\mathcal{C} = W \times \mathbb{N}_{|T|}$ the set of possible clues.

The Game module then passes c and U to the active team's guesser agent, which returns its guess v. That is, the guesser agent implements a function $\gamma : \mathcal{C} \to U$. Depending on the conditions described above, the Game module determines whether the active team may guess again, if the round has ended, or if the game is over. At the end of the game, the Game module outputs a result that indicates which team won and includes the board, the key, and the history of player actions for convenience in later analysis.

Spymaster AI

The spymaster's task is to choose an intended set $I \subseteq T$ and a clue word w. To play effectively, it must balance maximizing both $|I|$ and its estimated likelihood that w will induce a guess v such that $v \in I$. Playing very safely by giving clues that correspond to exactly one word card will likely lead to a low number of incorrect guesses but may progress too slowly to beat the opposing team. Conversely, choosing a clue that tries to represent too many of the team's word cards at once will likely lead to a vague clue that will confuse or mislead the spymaster's teammates.

Recall that guessing incorrectly ends that team's turn at best and at worst reveals one of the other team's target words for them or loses the game immediately by guessing the assassin. Thus, it is also helpful to consider the set $X = N \cup P \cup A$ of words with which the clue word w should *not* be associated.

Figure 3 gives a diagram of our spymaster AI agent that implements the function σ. It generates combinations of words to consider for I and passes each combination, along with X, into its word association module. That module (described below) returns a list of candidate clue words along with a confidence score for each. The agent selects the candidate with the best confidence as its clue word w. Our spymaster's procedure for choosing I relies on the heuristic that a clue representing between two and four words is a reasonable balance of reserved and aggressive play.[6]

In the case that only one of the team's word cards remains uncovered, it is trivial to select $I = T$. In all other cases, the spymaster AI will use the set $\mathcal{J} = \{J \in 2^T \mid 2 \leq |J| \leq 4\}$ to query a *word association module* (WAM), which computes a function $\alpha(\mathcal{J}, X) = (w, k) = c$. The spymaster then passes the returned clue c to the game module.

Word Association Module (WAM)

To implement α, the WAM uses the sets J, X to construct an abstract parameterized family of scoring functions $\sigma_{J,X} : W \to \mathbb{R}$ that maps a word $u \in W$[7] to a confidence score reflecting how well u is positively associated with the words in J *and* negatively associated with the words in X.

Given σ, we compute α as follows. For each $J \in \mathcal{J}$:

1. rank order W by the score $\sigma_{J,X}(u), \forall u \in W$
2. put the top m words into a candidate set C_J
3. compute $\mu_J = \frac{1}{|C|} \sum_{u \in C} \sigma_{J,X}(u)$

Next, find the set with the highest average confidence score, $J^* = \operatorname{argmax}_{J \in \mathcal{J}} \mu_J$ and, from its associated candidate words, C_{J^*}, find the word with the highest score, $w^* = \operatorname{argmax}_{u \in \mathcal{C}_{\mathcal{J}^*}} \sigma_{J,X}(u)$. Finally, return the tuple $(w, |J^*|)$.

By choosing the combination J^* with the highest *average* confidence, the model favors combinations that are more closely related altogether, even though another combination may have a single candidate word with higher confidence.

The primary language faculty required to play Codenames is knowledge of the relationships between words, both positive and negative. Knowing which words positively relate to each other is a necessary baseline skill, and understanding negative relationships between words is important for an agent to perform well.

For example, two Codenames word cards are "ambulance" and "doctor". These words are closely related, but if "ambulance" was one of a team's word cards (that is, "ambulance" $\in T$) and "doctor" was the other team's [or the assassin] ("doctor" $\in X$ or "doctor" $= a$), it would be important to exclude clue candidates that positively associate with "doctor". In that scenario, "siren" or "fast" would likely be a better clue than "emergency" or "injury".

Thus in this example, when $J \subseteq T$ contains "ambulance", we desire $\sigma_{J,X}(\text{“}siren\text{”}, k) > \sigma_{J,X}(\text{“}injury\text{”}, k)$.

[5] The notation 2^S is shorthand for the power set of S.

[6] As stated in the Codenames rulebook: "Getting four words with one clue is a big accomplishment."

[7] While this likely is technically incorrect, in the sense that any language model is likely subject to some out-of-vocabulary words, the language models used here support large enough vocabularies that they render the point basically moot—word2vec has a vocabulary size of 500K words, and GPT-2 uses a vocabulary of 50K sub-word tokens that likely translates to a functional word-level vocabulary even larger than that of word2vec.

The WAMs we experiment with are intended to serve as good players regardless of whether their teammates are human or other black-box AI players. Thus, a relevant consideration is whether the clues generated by the WAM are understandable to a broad audience. An obscure clue word and/or uncommon relationships to the word cards would likely confuse or mislead the spymaster's teammates. For this reason, statistical language models are a good fit for this task. We leverage them for both types of WAM.

We reiterate that the spymaster AI is designed to use *any* word association module that computes α, independent of how the WAM models language or how it uses the model to generate clues—the WAM can implement the function family σ in any way that maps words to scores. We implement three different versions: two that use a scoring function based on cosine similarity between word embeddings and one that uses conditional probabilities from autoregressive text generation by a language model.

Word Embedding WAMs

Both word2vec and GPT-2 feature word embeddings which are well suited to word relationship tasks because they allow semantic word comparisons using simple geometric, vector-based operations.

We built two word association modules that use word2vec and GPT-2 embeddings, respectively, to compute σ. Words are converted to real-valued embedding vectors using a language-model-specific embedding function $v : W \to \mathbb{R}^d$. For a word u, positive word set J and negative word set X, $\sigma_{J,X}$ is then computed using cosine similarity between the word vector $v(u)$ and a mean set vector $\mu_{J,X}$:

$$\sigma_{J,X}(u) = \frac{\mu_{J,X} \cdot v(u)}{||\mu_{J,X}|| \, ||v(u)||}$$

where

$$\mu_{J,X} = \frac{\sum_{v \in v(J)} v - \sum_{v \in v(X)} v}{|J| + |X|}$$

where we slightly abuse notation by overloading v to embed a set of words into a set of embedding vectors. Note that this embedding function is the only language-model-specific component of this approach—word2vec and GPT-2 learn their embedding spaces in different ways.

Text Generation WAM

Text generation is a powerful function of the GPT-2 language model. For our third implementation, we built a word association module that uses text generation to construct $\sigma_{J,X}$. To generate text, GPT-2 takes a prompt and generates tokens that are, according to its model, likely to follow. We designed prompts that state that a list of words related to an input word will follow, ending with a colon, followed by a comma-separated list of such words. The last line of the prompt ends after the colon, prompting GPT-2 to complete what comes next with an appropriate list. Here is an example of such a prompt; note that all three lines comprise a single prompt:

This is a list of words related to ambulance: paramedic, emergency, doctor.

This is a list of words related to boat: water, fish, captain.

This is a list of words related to school:

We experimented with three such prompt templates for use in the spymaster AIs. Each prompt asks GPT-2 to list words related to an input. The first prompt asks for words related to a single (positively associated) word. The second asks for words that are positively associated with two words. The third asks for words that are positively associated with one word and negatively associated with another. Because J and X can contain more than one or two words, we iterate over all possible template completions using words from J and X to construct the set C of possible generated completions. Let $\pi_i(Y)$ be the prompt created by adding the tuple of words Y to template type i and let $Z_i(Y)$ be the set of words generated by GPT-2 when prompted with $\pi_i(Y)$. Then, for the first template type (one positive association),

$$C = \bigcup_{u \in J} Z_1(u)$$

for the second template type (two positive associations),

$$C = \bigcup_{(u,v) \in J \times J} Z_2((u,v))$$

and for the third template type (one positive and one negative association),

$$C = \bigcup_{(u,v) \in J \times X} Z_3((u,v))$$

In the case that $|J| = 1$ or $X = \emptyset$, the module defaults to the single positive prompt template.

Each of these prompts is templated to allow for arbitrary input words, and the generated text is post-processed to extract only words in a comma-separated list. Any other output is discarded. The list is then filtered so that only valid, nonduplicate Codenames clues remain (e.g. single words that do not contain any of the word cards in U). The set of words that remains is included in the candidate set C.

For a word u, positive word set J and negative word set X, $\sigma_{J,X}$ is then computed using the generative model's conditional probability for u:

$$\sigma_{J,X} = p(u|\pi_i(Y)) \text{ for } Y \text{ a valid tuple from } J, X \text{ for type } i$$

Table 1 shows examples of each prompt template. The first and second columns list the template inputs: a single positive association, two positive associations, and one positive and one negative association. The third column shows the complete prompt with GPT-2's generated completion text. The input words are shown in italics, and the generated text in bold; all other text is the template. Note that any newlines are explicitly contained in the template or generated text. The final column shows the result of post-processing the generated text to extract valid clue candidates.

The GPT-2 model is not fine-tuned; its output relies solely on the prompt. The open-ended nature of text generation means that it is susceptible to noise in the output. We found that using a small number of template inputs reduced that

Positive	Negative	Templated Prompt + Generated Text	Candidates
cook		This is a list of words related to ambulance: paramedic, emergency, doctor. This is a list of words related to boat: water, fish, captain. This is a list of words related to *cook*: **urn, fire, vessel.**	urn, fire, vessel
hospital, spell		This is a list of words related to flag and state: country, government, county. This is a list of words related to mammoth and pyramid: ancient, large, heavy. This is a list of words related to bridge and skyscraper: concrete, blueprint, tall. This is a list of words related to *hospital* and *spell*: **crisis, catastrophe, crisis, disaster.**	crisis, catastrophe, disaster
lock	carrot	This is a list of words that are related to ambulance but not doctor: siren, engine, fast. This is a list of words that are related to bat but not duck: cave, night, fur. This is a list of words that are related to queen but not king: regina, woman, wife. This is a list of words that are related to *lock* but not *carrot*: **urn, house, castle, castle. This list is the closest of the**	urn, house, castle

Table 1: Examples of the prompt templates used in our three text generation word association modules. The positive and negative inputs are inserted into the templates which GPT-2 then uses to generate the bolded text. That text is post-processed to extract a list of valid candidate clues.

noise, which is reflected in our three templates. As such, both of the positive-only templates disregard X. We experimented with how well each template performed while making these trade-offs.

We also experimented with different wordings for the prompt templates, for example beginning each line with “These words are related to...” instead of “This is a list of words related to...” We discuss why we chose the wording and number of inputs for the final prompt templates in a later section.

Guesser AI

As discussed previously, the task of the spymaster’s teammates is to guess which word cards the clue is intended to represent. Therefore, the guesser agent is simpler, and the module requires only a word embedding model to calculate it. Figure 3 includes a diagram of our guesser AI. The agent uses its word similarity module to choose the word $u^* \in U$ that is most related to the clue $c = (w, k)$, again using cosine similarity:

$$u^* = \underset{u \in U}{\operatorname{argmax}} \frac{v(u) \cdot v(w)}{||v(u)||\,||v(w)||}$$

We note that this guessing process disregards the number k provided in the clue. While that is additional information that the guesser could leverage, we believe that this simplified approach is sufficient for the research task at hand, namely the exploration of different language models as creative Codenames players. From this perspective, the spymaster is the more interesting agent and was therefore the focus of our experiments. Furthermore, whatever information the clue number provides is supplementary to the associations between the clue word and the word cards. At most, it could be used to refine the language module’s association scores.

GPT-2’s text generation function is open-ended; it can generate any tokens that appeared in its training corpora. Therefore, the likelihood of the model generating the specific words found on the board is very low. We experimented with building prompts for the guessing task. For example (again, all four lines comprise one prompt):

> Which of the words ambulance, shoe, and Moscow is most closely related to siren? ambulance
>
> Which of the words chick, China, and bolt is most closely related to lightning? bolt
>
> Which of the words opera, casino, pilot is most closely related to fancy? opera
>
> Which of the words India, needle, shop is most closely related to sharp?

In this example, the intended result is that GPT-2 generates “needle” as the next word. We tested whether the intended word appeared at all before the first generated newline character. Our experiments showed that GPT-2 generated the intended word in less than 5% of trials. We therefore did not build a GPT-2 text generation guesser at this time.

Model Comparison

As discussed above, the word association tasks that are required to play Codenames, especially in the spymaster role, provide opportunities for creativity. Each of our player agents includes a pluggable, language model-driven module that serves as the creative heart of its playing procedure. By comparing these modules within the well-defined creative space of a competitive language game, we can concretely reason about their performance.

To make these comparisons, we built a lightweight test harness that plays games of Codenames between two teams of agents. These games are played using the same list of word cards available in the retail game. Each team consists of a spymaster agent and a guesser agent who are agnostic to the implementation of the agents they are playing with and against. Codenames can be played with a small team of guessers collaborating to guess their spymaster’s clues, but teamwork between guesser agents is outside the scope of this work. The fundamental task of testing language models in this game setting can be adequately explored with a solo guesser.

To provide benchmarks for the agents' performance, we implemented simple guesser agents that guess randomly or cheat. The random guesser agent serves as a lower bound on acceptable performance for an AI agent. The cheat guesser agent simply guesses n correct words each round, then passes. This serves as a rough but easy-to-compute pace against which to compare each agent.[8] A benchmark team consists of either a random or cheat guesser agent paired with a trivial spymaster agent that returns a dummy clue that the guesser disregards.

We created teams out of every pairing of spymaster and guesser AI agents, regardless of their underlying language models. These teams were played against one another and the benchmark agents, and their win/loss ratios were recorded. The name of each team is given as "[spymaster]4[guesser]", meaning the spymaster is making clues fo(u)r the guesser. "w2v" stands for word2vec, "gpte" stands for GPT-2 embedding model, and "gptp" stands for GPT-2 prompt (text generation) model. The six teams were w2v4w2v, w2v4gpte, gpte4w2v, gpte4gpte, gptp4w2v, and gptp4gpte. Each team played 30 games against every other team and 30 games against a random team, a cheat team with $n = 1$, and a cheat team with $n = 2$.

Results

In this section, we report the results of our experiments. Our primary objective is to demonstrate that a competitive language game task allows for quantifiable comparison between agents. By powering our player agents with language association modules, we show by extension how such modules can be evaluated with concrete performance metrics.

We ran experiments using three spymaster agents and two guesser agents, in addition to the cheat and random benchmark agents. The two guesser agents were built on word2vec and GPT-2 word embeddings. The spymasters used word2vec word embeddings, GPT-2 word embeddings, and GPT-2 text generation, respectively, to perform word association tasks. The text generation spymaster could further be configured to use one of the three prompt templates shown in Table 1.

GPT-2 Prompt Comparison

We compared the performance of the three prompt templates by playing text generation agents (paired with both guessers) against cheat benchmarks with $n = 1$ and $n = 2$ as well as the random player. Table 2 shows the results of playing 10 games between those teams. Each template performed similarly against the random player, with most teams being able to beat it consistently. Similarly, all teams performed uniformly poorly against the cheat benchmarks. In the tests that follow, we used the template with one positive input as the prompt for the GPT-2 text generation module.

Comparing Agent Teams

By playing our various teams of agents against one another, we can judge their relative performance at the word association task. The benchmark cheat and random agents provide a more objective performance measure. Figure 4 shows the win/loss ratio for each combination of agents playing 30 games against every other team and the benchmark agents. A darker color indicates a higher win rate for the team on the y-axis versus the team on the x-axis. Recall that we did not implement a GPT-2 text generation guesser. We also did not play a team of agents against a team of the same agents.

Prompt	Guess	Cheat 1	Cheat 2	Rand
1 positive	w2v	0-10	0-10	6-4
	GPT-2	1-9	0-10	7-3
2 positive	w2v	0-10	0-10	7-3
	GPT-2	0-10	0-10	5-5
pos + neg	w2v	0-10	0-10	9-1
	GPT-2	0-10	0-10	4-6

Table 2: Win-loss ratio results of playing different text generation spymasters against benchmark agents, each using one of the three prompt templates.

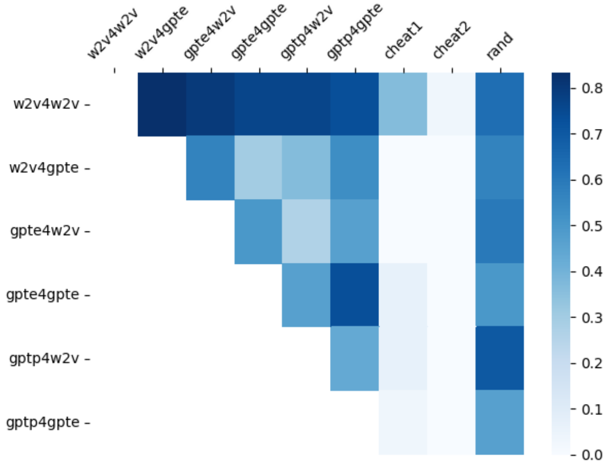


Figure 4: Heatmap of win/loss ratios after 30 games for each team of Codenames agents playing against one another and the unintelligent benchmark agents.

Looking at the results, we can see that the word2vec spymaster and guesser team performed best overall. Conversely, the team of the GPT-2 text generation ("gptp") spymaster and GPT-2 embedding guesser was the weakest. Most teams beat the random agent at least half of the time, with gpte4gpte and gptp4gpte teams losing as often as they won against it. None of the teams could consistently win against the cheat agents, but the word2vec spymaster/guesser team was able to beat the cheat with $n = 1$ about a third of the time.

Finally, we note that these tests are automated and can be carried out on any new or modified agent to test its performance at the creative spymaster task under the same circumstances. This will allow for easy evaluation and comparison

[8] Anecdotally, it seems a human team would find a cheat agent with $n = 3$ challenging and struggle to ever beat one with $n = 4$.

as improved models are developed in the future.

Discussion

Our experiments with Codenames AI agents serve a dual purpose: to present an initial attempt at designing agents to play the game; and (more importantly) to demonstrate that a competitive language game is a creative domain with a unique capability for evaluating agents.

Our Codenames Agents

It is somewhat surprising at first blush that word2vec outperformed GPT-2 word embeddings at playing Codenames. This may be attributable to differences in how the two models are trained. Word2vec's skip-gram and negative sampling model is trained under circumstances that are very similar to the task of finding words associated with an arbitrary set of positive and negative concepts. GPT-2, while not unsuited for the task at hand, is trained to more generally minimize cross entropy in its language model. Perhaps training or fine-tuning a transformer module using skip-grams and negative sampling would bring their power to bear on this more specific task.

This surprising result was demonstrated very clearly by the test methodology of playing a competitive game with the two models. This serves as another example of how this creative task is useful to CC research. Additionally, improved future spymaster agents can be tested against these same models to evaluate their performance.

Designing a word association module using GPT-2 text generation relied heavily on prompt engineering. Prompting the model with examples in the form of a comma-separated list resulted in the generated text taking a similar form. This allowed for consistent input to the post-processor to extract clue candidates.

More challenging was engineering a prompt template that harnessed the power of the language model to generate high-quality word associations. As described previously, we settled on three prompt templates that sought associations with one word, two words, and one positive and one negative word. By contrast, the word embedding models calculated word associations using an arbitrary number of positive and negative word embedding vectors.

We found that increasing the number of input words in the template tended to increase the noise of the generated text without improving the quality of its associations. However, the results reported in the previous section show that the GPT-2 text generation module prompted with one positive input performed about as well as the GPT-2 word embedding model.

The Future of Codenames as a Creative Task

Successfully playing Codenames requires robust knowledge of relationships between words, but the input and output for player agents are single words or lists of words. This stands in contrast with a game like Werewolf which requires more complete communication skills as players attempt to figure out hidden roles. There is a smaller conceptual distance between the language model and the agent's performance playing Codenames.

Playing a competitive game allows for automated and easy-to-compare metrics for modules with open-ended tasks, such as using GPT-2 text generation to compute word associations. Further, by first building a Codenames test harness, we were able to quickly test and compare prompts. For example, we found that prompts beginning with "This is a list of words related to..." gave better results than those beginning with "These words are related to...".

The nature of competitive language games like Codenames allows for future improved and novel agents to be tested under identical conditions to the ones presented here. We foresee an improving field of creative Codenames agents that can be tested automatically against one another.

Bodily and Ventura present an argument for increased social consciousness of CC systems, especially as they eclipse human performance (2020). This is largely motivated by the triumph of AlphaGo over a top-ranked human player at Go, which is a creative task with a "well-defined and universally-recognized way of comparing" performance.

Codenames does not have the same depth, history, or audience that Go has, but it is quite popular in its own sphere. It shares a similar potential for creativity but operates in the domain of language. Creativity in language domains is a valuable and well-studied aspect of computational creativity, and Codenames could serve as a test bed to develop creative language modules that could be exported for use in those more traditional domains.

These arguments for Codenames as a valid and useful creative domain apply to other competitive language games as well. We encourage the research community to seek out and experiment with such games as well-defined creative tasks.

Conclusion

A common difficulty in systematizing creativity is identifying an accurate and concrete goal. High-quality creative output is difficult to quantify, and abstractions or estimations are usually required. We argue that competitive language games such as Codenames are a useful creative domain because they feature well-defined win and lose states while still allowing for creative expression.

We present a test framework for playing games of Codenames between AI agents both to describe a new creative system and to demonstrate the efficacy of the domain itself. This framework is modular to allow for any player agent to be evaluated and includes benchmark agents to provide more objective performance metrics.

Each creative domain provides unique challenges and new perspectives on what creativity is, how to reason about it, and what tools facilitate computational creativity. Adding competitive language games to CC's suite of canonical creative domains will allow for more rigorous evaluation and comparison of its creative systems. There is more to creativity than winning a game, but in the face of a dearth of concrete measures of creative performance, competitive language games can serve as a valuable proxy for such evaluation.

Author Contributions

Both authors planned and designed the system, B.S. wrote the code and ran experiments, and both authors contributed to the writing.

Acknowledgements

None.

References

Bach, S. H.; Sanh, V.; Yong, Z.-X.; Webson, A.; Raffel, C.; Nayak, N. V.; Sharma, A.; Kim, T.; Bari, M. S.; Fevry, T.; Alyafeai, Z.; Dey, M.; Santilli, A.; Sun, Z.; Ben-David, S.; Xu, C.; Chhablani, G.; Wang, H.; Fries, J. A.; Al-shaibani, M. S.; Sharma, S.; Thakker, U.; Almubarak, K.; Tang, X.; Tang, X.; Jiang, M. T.-J.; and Rush, A. M. 2022. Promptsource: An integrated development environment and repository for natural language prompts. arXiv:2202.01279.

Bodily, P., and Ventura, D. 2020. What happens when a computer joins the group? In *Proceedings of the 11th International Conference on Computational Creativity,*, 41–48.

Chvátil, V. 2015. *Codenames*. Kladno, Czech Republic: Czech Games Edition. Board Game.

Dorst, K. 2011. The core of 'design thinking'and its application. *Design studies* 32(6):521–532.

Guilford, J. P. 1956. The structure of intellect. *Psychological bulletin* 53(4):267.

Harris, Z. S. 1954. Distributional structure. *Word* 10(2-3):146–162.

Jolin, D. 2016. The rise and rise of tabletop gaming. *The Guardian*. Accessed: 2020-10-05.

Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; and Neubig, G. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. arXiv:2107.13586.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013a. Efficient estimation of word representations in vector space. *arXiv* abs/1301.3781.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26, 3111–3119.

Nevskiy, O., and Sidorenko, O. 2015. *Mysterium*. Paris, France: Libellud. Board Game.

Ohlsson, S. 1992. Information-processing explanations of insight and related phenomena. *Advances in the Psychology of Thinking* 1:1–44.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners. `https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf`.

Řehůřek, R., and Sojka, P. 2010. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 45–50. Valletta, Malta: ELRA.

Sahlgren, M. 2008. The distributional hypothesis. *Italian Journal of Disability Studies* 20:33–53.

Ushan, A. 2017. *Spyfall 2*. Moscow, Russia: Hobby World. Board Game.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 31, 6000–6010.

Veale, T. 2006. Re-representation and creative analogy: A lexico-semantic perspective. *New Generation Computing* 24(3):223–240.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; Davison, J.; Shleifer, S.; von Platen, P.; Ma, C.; Jernite, Y.; Plu, J.; Xu, C.; Scao, T. L.; Gugger, S.; Drame, M.; Lhoest, Q.; and Rush, A. M. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45. Online: Association for Computational Linguistics.

A Game of Essence and Serendipity: Superb Owls vs. Whisking Woodpeckers

Guendalina Righetti
Free University of
Bozen-Bolzano
Bolzano, Italy
guendalina.righetti@stud-inf.unibz.it

Oliver Kutz
Free University of
Bozen-Bolzano
Bolzano, Italy
oliver.kutz@unibz.it

Daniele Porello
University Of Genoa
Genova, Italy
daniele.porello@unige.it

Nicolas Troquard
Free University of
Bozen-Bolzano
Bolzano, Italy
nicolas.troquard@unibz.it

Abstract

The representation of everyday concepts is important for a number of applications, ranging from the Semantic Web to NLP and general AI. We propose here a detailed case study of the *Leuven concept database* (LCD), which is a rich database of commonsense knowledge, written in natural language. We aim to convert the commonsense knowledge contained in the LCD into a format suitable for implementation and practical application. We then investigate a hybrid approach that combines a syntactic analysis of the surface structure of the LCD entries with a semantic and ontological analysis of those entries, considering also the role of other cognitively-grounded facets of core knowledge. The approach therefore suggests a systematic portfolio of disambiguation modes with the goal of improving the match between everyday meaning of concepts and formal semantics. Finally, we illustrate the practical usefulness of this approach in a concrete computational implementation for concept combination.

Introduction

Commonsense knowledge and specifically the representation of everyday concepts is a crucial ingredient in many applications, ranging from the Semantic Web to NLP and general AI. The word "commonsense" groups different aspects of human knowledge, which permeate our experience of the world and allow us to move therein. Commonsense knowledge includes our ability to distinguish between single objects and classes of objects, to distinguish between animate and inanimate things, but also more mundane knowledge: the fact that fish live only in water (and normally do not have a job), the fact that vehicles need fuel, or the fact that my dad is necessarily born before me. Commonsense knowledge is acquired by humans through experience and throughout life in an almost completely effortless way. Despite the long tradition of research (McCarthy, 1959; Lenat, 1995) investigating how to bring this kind of knowledge from human to machine, it is still a wide-open research question. At the same time, any progress in this field directly benefits a number of AI applications.

As a case in point, in the context of Computational Creativity the representation of commonsense knowledge is crucial when dealing with Computational Conceptual Blending. In Cognitive Linguistics, Conceptual Blending has been proposed as a general cognitive process underlying, among others, the human ability to creatively integrate and combine concepts (Boden, 1998; Fauconnier and Turner, 1998). Accordingly, a blend is constructed by selectively mapping the shared features of different (mental) input spaces into a generic, shared, mental space. The blend develops then its own emergent structure, which derives from the combination of the projected features. For humans, this process may happen imperceptibly, by exploiting information they possess, specifically relying on their commonsense knowledge. Arguably, some of the most interesting blends originate from the resolution of clashes stemming from the commonsense information which is coded into, and sometimes hidden in, the input concepts.

Computational Conceptual Blending (CCB) aims at formally interpreting and capturing the process of conceptual blending and integration. Different, though related, frameworks have been proposed in the literature, either to formally model or to replicate the process of conceptual blending (Eppe et al., 2018; Neuhaus et al., 2014; Veale, 2019; Ontañón and Plaza, 2010; Hedblom, Righetti, and Kutz, 2021; Gonçalves, Martins, and Cardoso, 2017). Rather obviously, computational systems are forced to reason with the information they are presented with, and to bootstrap the clashes and the blending process in general, CCB systems need commonsense knowledge to be represented in the input spaces. Beyond the study of the heuristics involved in the computational process of creatively blending concepts, it is then also worth focusing on the formalisation of the commonsense information which is needed as a propellant to steer the whole process.

To this end, we focus here on a detailed case study of the Leuven concept database (De Deyne et al., 2008; Ruts et al., 2004), which is used as a source of commonsense knowledge to be converted into a format which will be then suitable for practical application. The Leuven concept database (LCD) contains information, gathered by a group of psychologists at the University of Leuven, over the features exhibited by 15 category labels (here often referred to simply as *concepts*), and provides evidence on human conceptualisation. The conceptualisations that emerge from the LCD do not necessarily reflect a *good* definition of the concepts involved—at least not in a normative sense. It aims at being a good description of what people have in mind when they think about those concepts, and of the meaning they associate with them. Therefore, the database is permeated by "commonsense information", and exhibits some of the basic ambiguities related to the use of natural language. These conceptualisations, therefore, constitute an excellent point of observation on the challenges to be faced to make this information machine interpretable. We propose here a study which addresses exactly these difficulties. In order to make the content of the LCD available for practical application, a process of formalisation is needed: we exploit here the Web Ontology Language (OWL) as a prominent starting point. Being a computational, logic-based language, OWL obviously imposes certain limitations in terms of expressivity. The translation from the LCD to OWL thus involves a trade-off between the language's expressive power and the desire to preserve as much information as possible. Another boundary in the translation is set by the presence of background foundational ontology distinctions, which are used to inform some of the formalisation choices in the process of translation. In particular, we argue, and present some examples, that exploiting deep ontological distinctions enables us to impose order and coherence (when possible) to the information in the LCD, helping also to disambiguate some of the hidden meaning within the data.

Finally, we conclude the paper illustrating the practical usefulness of this approach in a concrete computational implementation for concept combination. This will here serve as a demonstration and a possible use of the resulting formalised commonsense knowledge.

Related Work

Many practical AI applications require complex inferences, which, in turn, require large common-sense knowledge bases. Typical example are chatbots, or domotic applications, e.g. involving 'intelligent' cooking or cleaning assistants, which need to navigate human spaces with a sufficient level of the involved common sense inferences Krieg-Brückner et al. (2015); Bateman et al. (2018). In practice, this need has often resulted in the use of structured lexical databases, semantic networks, or linked data, such as WordNet (Fellbaum, 2005), ConceptNet (Speer, Chin, and Havasi, 2017) and DBpedia (Auer et al., 2007) as a link between natural language and higher level semantic representations. Despite their usefulness, these repositories often show some level of ambiguity, which demonstrates the lack of a common agreement on the meaning of the lexical entries. In order to overcome this difficulty, a number of works have proposed different approaches to provide these databases deeper semantic support (Fellbaum and Hicks, 2019; Silva, Freitas, and Handschuh, 2016; Gangemi et al., 2012; Schmidt et al., 2019; Gangemi et al., 2003). The key ideas behind those approaches is to make these repository "ontology-like", as far as possible.

In order to achieve this level of formalisation, many of the approaches mentioned above appeal to foundational ontology (FO — such as BFO, DOLCE, GFO, SUMO, etc.) which provide a common vocabulary through imposing fundamental ontological distinctions. In (Gangemi et al., 2003), for example, a connection is drawn between WordNet's upper level synsets and the foundational ontology DOLCE, and, more recently, (Silva, Freitas, and Handschuh, 2016) enlarged that alignment in order to include also verbs. In (Schmidt et al., 2019), a complete manual alignment between WordNet and a different Upper Ontology (SUMO) is proposed. Continuing that tradition, (Gangemi et al., 2012) propose a tool for automatically typing DBpedia entities, which relies on the alignment to both Wordnet supersenses and a subset of DOLCE Ultra Lite classes. Crucially, these works often use a top-down approach which propagates certain top level distinctions of the foundational ontology onto the more general entries in the database at hand, exploiting its internal relation (e.g. the hyponym relation).

We follow here a related but different strategy, based on a detailed case study of the Leuven Concept Database De Deyne et al. (2008); Ruts et al. (2004). Instead of assuming a specific FO and propagating its distinction through the database, we exploit the inverse, bottom-up, direction. We analyse the intended meaning of the information contained in the LCD and individuate seven *modes of disambiguations*, i.e. seven high level distinctions, ranging between ontological and cognitively relevant ones, which implicitly underlie the content of the LCD. Once individuated, these distinctions steer the analysis of the database, and thus the rendering choices of our translation into OWL.

We carried out the translation into OWL manually. There exists different tools for automatic natural language to OWL translation (Völker, Hitzler, and Cimiano, 2007; Emani et al., 2019; Draicchio et al., 2013; Nguyen, Razniewski, and Weikum, 2021). In order to be effective, these tools require very clear assertions and showing a regular structure. In contrast, the commonsense features collected in the Leuven concept database, in most cases, do not show this kind of regularity and lack of ambiguity that these tools presuppose.

The Leuven Concept Database

Data gathering

The Leuven concept database[1] is a large-scale data set that associates sets of features both to concepts (or categories' labels, e.g. *Bird*) and to exemplars (or lexical entries, e.g. *magpie*). The data collection was carried out by the ConCat group at the University of Leuven from 2004 to 2008 (Ruts et al., 2004; De Deyne et al., 2008), and it consists of 15 categories and 420 associated exemplars. More precisely, the data set covers the domain of animals (**birds**, **fish**, **insects**, **mammals**, **reptiles** together with **amphibians**, with an average of 25 exemplars for each category label, and a total of 131 exemplars), and it collects information on the artifact domain (**musical instruments**, **tools**, **vehicles**, **clothing**, **kitchen utensils**, **weapons**, for a total of 169 exemplars over the six categories), on **fruit** and **vegetables** (for a total of 60 exemplars) and activities (**professions** and **sports**, again for 60 exemplars). At least a thousand students were involved in the experiments. All the material was collected in Dutch, but also an English translation is provided to make

[1] Available at `https://simondedeyne.me/data`

the data available for further experimental and modelling approaches.

The studies conducted at the University of Leuven are placed in the debate between the Prototype Theory and the Exemplar Theory (Storms, De Boeck, and Ruts, 2000), and therefore present a series of experiments that aim to investigate aspects of one or the other theory. We are here mostly interested in the studies pertaining to a *feature-generation task*, where subjects were asked to provide lists of features in relation to the 15 category labels presented in **bold** above.

Participants' responses to the feature generation task were manually aggregated and adjusted with minimal stemming. Information was retained on the features' production frequency, which can be considered an indirect measure of their importance. Further, the importance of the features was also directly assessed by asking the participants to explicitly rate the importance of each feature in the definition of the concept for which they were previously generated (De Deyne et al., 2008). Figure 1 shows an example of the features generated for the category label *Bird* (see before the column MEAN). The table displays the features in Dutch and their translation to English. The numbers displayed in the table correspond to the importance ratings assigned to each feature by the participants of the experiments. The rating scale ranged from $+3$ (very important feature) to -3 (very unimportant feature). Globally, the feature generation task for the category label produced 28 features for the concept *Bird*.

Large-scale data-sets analysing the features exhibited by different concepts are quite rare in the literature, even though the possibility of using Amazon Mechanical Turk has made them more frequent (Vinson and Vigliocco, 2008; Buchanan, Valentine, and Maxwell, 2019). The LCD, however, shows some peculiarities that make it particularly suitable for our analysis: not only is it organised in an easily reusable shape, which makes it useful from a practical point of view; it also contains information about the importance of the features collected, which makes it interesting from a theoretical perspective, as will be explained in what follows.

The Leuven Concept-bundle

As it can be seen in the table for the concept *Bird*, the features collected relate to different aspects of birds, that range from habits ("builds nests", "eats worms"), to body parts or shape ("has wings", "has a beak"), to abilities ("can fly", "sings"), but that also pertain to more general cultural information (e.g, "is sometime kept as a pet", "is sometimes eaten by man"). A similar situation occurs for each of the concepts analysed, and if one takes a step back and looks at the features contained in the Leuven concept database as a whole, a rather fascinating picture emerges. Different features reflect different facets of the concepts involved, that in some cases barely stand together in the same description. For instance, some of the features of the concept *Fish* ("breathes under water", "has gills", "lays eggs", "lives in the sea") suggest a quite general definition of *Fish*, which relates to a somehow biological perspective on the concept. Other features describe instead the concept *Fish* in its relation with humans beings—and maybe with some subject's personal experience: some of them ("swims in aquarium", "is sometimes kept as a pet") focus on the pet dimension of *Fish*, while others ("contains omega3", "is tasty", "sometimes smells") relate to the food dimension. Also, features pertaining to different dimensions may be considered conflicting—at least at some level: does the fish live in the sea or in the aquarium? Is it a pet or is it tasty? Similar considerations apply to all the concepts in the database: a *Sport* is a hobby, is relaxing and is fun, but can also be a *Profession*, which in turn is defined as a source of stress and frustration (but also an activity which is advantageous for the society and the economy). *Clothes* protect against the cold, but can be a status symbol, they protect from the rain but express your personality; a *Tool* is an aid, but you can injure someone with it; *Vehicles* are polluting, but they may be environmentally friendly. This gives us an idea of the context sensitivity of everyday concepts (Yeh and Barsalou, 2006), but reveals also their polysemy—the fact that those category labels are used as umbrella words for slightly different meanings. This also recalls what Hofstadter (2001) called the process of chunking: the idea that humans build their concepts by gluing several concepts together through their lifetime, so that at the end a concept results in "nothing but a tightly packaged bundle of analogies".

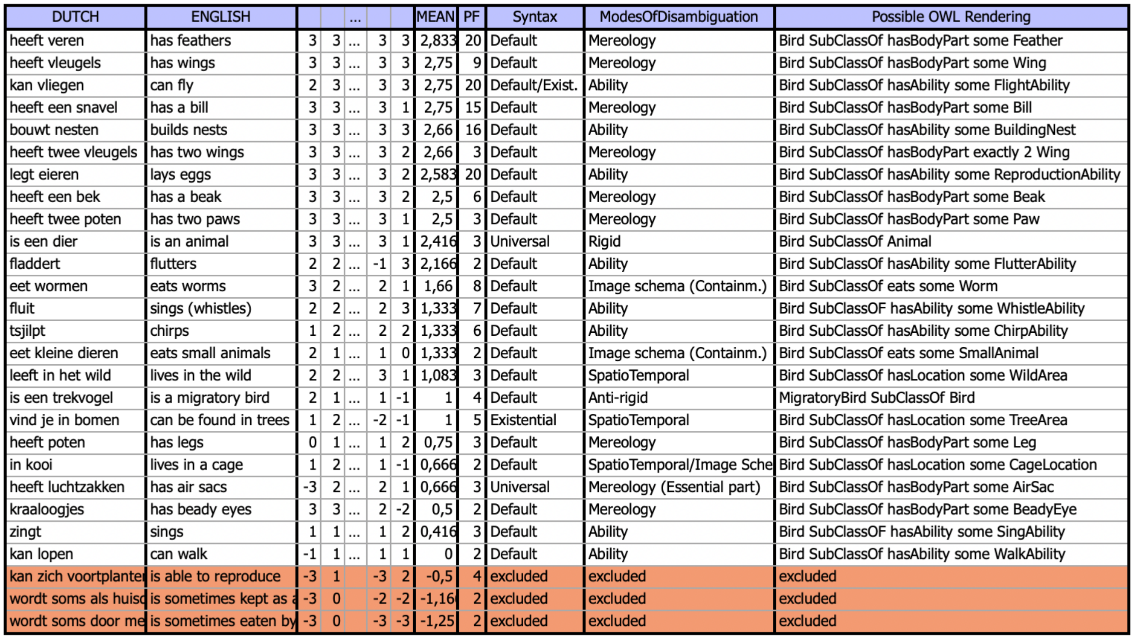
Taking a step forward and looking at the features more closely in terms of formalisation possibilities, some problems quickly emerge, e.g. regarding precision. One of the problems may be summarised as a lack of implicit knowledge. This does not only refer to the lack of fundamental categorical distinctions (see below), but also to the omission of some of the things that subjects may have considered obvious during the experiment. Subjects tend to omit some of the most obvious features (e.g. that a fish "has two eyes"), trying to focus on the more distinguishing ones (De Deyne et al., 2008). Also, they fail to specify some underlying knowledge, which they may consider not necessary for general understanding—fish are said to "swim in aquarium", compressing the more detailed information "some fishes swim in water contained in some aquarium". Another problem is the presence of errors within the data, most of which correspond to a naïve use of the "is-a" relation: a *Fish* is said to be a shark, a *Tool* is a hammer, etc.

From the LCD to OWL Ontologies

Interpreting the features

The problems described in the section above would not cause any issue for human understanding, which shows great flexibility in interpreting natural language sentences. However, when the goal is to make the features machine interpretable, they require some adjustments. Let us, for instance, consider to translate 'naïvely' the feature "swims in aquarium" into an OWL axiom, and to add it to an ontology of *Fish*. In the ontology there could be a definition of *swims*, maybe as an action which is performed only in a particular environment—namely in water[2]. In order to avoid inconsistencies, such as the identification of 'aquarium' and 'water', one may need then to fully specify the meaning and function

[2] Unless one wants to consider a metaphorical use of the word *swim*, which would make the situation even more complicated.



DUTCH	ENGLISH			...			MEAN	PF	Syntax	ModesOfDisambiguation	Possible OWL Rendering
heeft veren	has feathers	3	3	...	3	3	2,833	20	Default	Mereology	Bird SubClassOf hasBodyPart some Feather
heeft vleugels	has wings	3	3	...	3	3	2,75	9	Default	Mereology	Bird SubClassOf hasBodyPart some Wing
kan vliegen	can fly	2	3	...	3	3	2,75	20	Default/Exist.	Ability	Bird SubClassOf hasAbility some FlightAbility
heeft een snavel	has a bill	3	3	...	3	1	2,75	15	Default	Mereology	Bird SubClassOf hasBodyPart some Bill
bouwt nesten	builds nests	3	3	...	3	3	2,66	16	Default	Ability	Bird SubClassOf hasAbility some BuildingNest
heeft twee vleugels	has two wings	3	3	...	3	2	2,66	3	Default	Mereology	Bird SubClassOf hasBodyPart exactly 2 Wing
legt eieren	lays eggs	3	3	...	3	2	2,583	20	Default	Ability	Bird SubClassOf hasAbility some ReproductionAbility
heeft een bek	has a beak	3	3	...	3	2	2,5	6	Default	Mereology	Bird SubClassOf hasBodyPart some Beak
heeft twee poten	has two paws	3	3	...	3	1	2,5	3	Default	Mereology	Bird SubClassOf hasBodyPart some Paw
is een dier	is an animal	3	3	...	3	1	2,416	3	Universal	Rigid	Bird SubClassOf Animal
fladdert	flutters	2	2	...	-1	3	2,166	2	Default	Ability	Bird SubClassOf hasAbility some FlutterAbility
eet wormen	eats worms	3	2	...	2	1	1,66	8	Default	Image schema (Containm.)	Bird SubClassOf eats some Worm
fluit	sings (whistles)	2	2	...	2	3	1,333	7	Default	Ability	Bird SubClassOF hasAbility some WhistleAbility
tsjilpt	chirps	1	2	...	2	2	1,333	6	Default	Ability	Bird SubClassOf hasAbility some ChirpAbility
eet kleine dieren	eats small animals	2	1	...	1	0	1,333	2	Default	Image schema (Containm.)	Bird SubClassOf eats some SmallAnimal
leeft in het wild	lives in the wild	2	2	...	3	1	1,083	3	Default	SpatioTemporal	Bird SubClassOf hasLocation some WildArea
is een trekvogel	is a migratory bird	2	1	...	1	-1	1	4	Default	Anti-rigid	MigratoryBird SubClassOf Bird
vind je in bomen	can be found in trees	1	2	...	-2	-1	1	5	Existential	SpatioTemporal	Bird SubClassOf hasLocation some TreeArea
heeft poten	has legs	0	1	...	1	2	0,75	3	Default	Mereology	Bird SubClassOf hasBodyPart some Leg
in kooi	lives in a cage	1	2	...	1	-1	0,666	2	Default	SpatioTemporal/Image Sche	Bird SubClassOf hasLocation some CageLocation
heeft luchtzakken	has air sacs	-3	2	...	2	1	0,666	3	Universal	Mereology (Essential part)	Bird SubClassOf hasBodyPart some AirSac
kraaloogjes	has beady eyes	3	3	...	2	-2	0,5	2	Default	Mereology	Bird SubClassOf hasBodyPart some BeadyEye
zingt	sings	1	1	...	1	2	0,416	3	Default	Ability	Bird SubClassOF hasAbility some SingAbility
kan lopen	can walk	-1	1	...	1	1	0	2	Default	Ability	Bird SubClassOf hasAbility some WalkAbility
kan zich voortplanter	is able to reproduce	-3	1		-3	2	-0,5	4	excluded	excluded	excluded
wordt soms als huisd	is sometimes kept as a	-3	0		-2	-2	-1,16	2	excluded	excluded	excluded
wordt soms door me	is sometimes eaten by	-3	0		-3	-3	-1,25	2	excluded	excluded	excluded

Figure 1: **Bird**: an example of a "Feature via Category Label" table, plus annotations from our analysis. MEAN refers to the mean of the importance associated by the subjects to the features. PF is the production frequency.

of the object 'aquarium', and the image-schematic relation of containment it bears with the water.

These considerations suggest the need of a preprocessing phase, which we conducted at different levels. At the very first level, this preprocessing phase was carried out following the Gricean communication principle, or *Cooperative Principle*. In the context of any language exchange, the principle prescribes to "make your contribution such as is required (. . .) by the accepted purpose or direction of the talk exchange in which you are engaged" (Grice, 12 Dec 1975). In this specific case, the *purpose of the talk exchange* was the definition of the concepts proposed by the psychologists, and the *talk exchange* was more precisely the experiment completed by the participants. According to Grice, the violations of this principle, which here includes errors and imprecisions, should be interpreted in such a way as to protect the rationality of the speaker, according to Quine's *Principle of Charity* (Quine, 1960)), which prescribes interpreting a speaker's statements in the most rational way possible, and considering its best, strongest possible interpretation[3].

Despite these premises, some of the features produced in the Leuven experiments were difficult to interpret in the context of the definition of the concept—sometimes because blatantly false when stated for the whole class, sometimes because they were related to a semantic context completely different to the one proposed in the experiment. To give a taste: a *Fish* is "a constellation" and *Weapons* are "used in sport". Some of the features, then, captured biases of our language (and our society): a *Profession* "is different for men and women", and a *Kitchen Utensil* "is especially used by women", but a *Tool* "is primarily used by men". As described in more detail above, all the features generated in the experiments were afterwards judged in order to evaluate their applicability to the class at hand (see again the numbers in Figure 1). Inspired by the prototype-theoretic notion of *salience* (Rosch, 1973), and in order to exclude some of the most controversial features, we calculated the mean of subjects' judgments, and excluded the entries strictly below the threshold 0. This procedure allowed us to exclude 102 features, namely around 20% of the features.

The formalisation step

After the preprocessing step, the features are translated into OWL axioms. The Web Ontology Language (OWL) is one of the most widespread language for authoring ontologies. It allows the users to write explicit and formal conceptualisations of a domain model. We will just sketch here the features of the language, the interested reader may refer to (Antoniou and van Harmelen, 2009; McGuinness and van Harmelen, 2004) for a more in depth description.

In particular, OWL is a logic-based language: it is mapped to Description Logics, i.e. decidable fragments of first-order logic. This provides OWL with a clear, well defined, formal semantics and efficient reasoning services. The reasoning support is important not only to compute ontologies' implicit knowledge (i.e. the entailed statements), and thus to reason over the axioms, but also to check their consistency, the presence of unintended consequences, etc. At the same time, efficient reasoning services require some limitations in the expressiveness of the language. Some trade-off is then necessary between the performance of the reasoning and the

[3] Interestingly enough, Quine developed this principle in the context of language translation.

language's expressive power, which should allow the user to express large volumes of knowledge.

More precisely, OWL allows to express knowledge about classes, instances and binary relations between instances. It provides different constructs to declare the different entities of the language: here we mainly deal with the constructs class, object property and individual. A class defines a set of individuals that share some properties; object properties are used instead to assert binary relationships between individuals; individuals are instances of the classes. For example, we may want to declare the class of *Bird* as the set of those instances that share the features described above. All the 15 categories described in the Leuven concept database are indeed examples of classes. If we want to populate the class, we may declare Tweety as an individual of the class Bird. Consider instead the feature "builds nest": the word 'builds' should be interpreted as an object property, which relates the instances of the class *Bird* and the instances of the class *Nest*. At the same time, the set of all entities that build nests provides another example of a class, which *Bird* is a subclass of. Classes can indeed be organised in hierarchies, according to their generality, by means of the "subClassOf" relation, which behaves like the subsumption relation in Description Logic. We may also declare that two classes are "disjoint", having no common instance, and that two classes are "equivalent", having exactly the same instances.

The semantics of the "subClassOf" relation implies that all the elements of the sub-class are also elements of the super-class (it is indeed the subset relation). Asserting that *Bird* is a subclass of the class of entities which *build Nest*, means then that all the instances in the class of Birds build nests, without exceptions. Obviously this is a quite strong requirement when we are dealing with natural language formalisation and everyday concepts. Some of the features are described by people by means of expressions which emphasise their partial applicability to the class into consideration (e.g. sometimes, can have, etc). In other cases, this is implicit in the use of everyday language (e.g. people may assert that birds can fly, but this does not imply that they believe that a penguin is not a bird). Also for this reason there has been some work recently trying to allow a more cognitively grounded modelling (Porello et al., 2019; Righetti et al., 2019, 2021a), as well as defeasible subsumption (Britz and Varzinczak, 2017; Casini and Straccia, 2010), which allows to handle exceptions and counterexamples.

Following these intuitions, the features collected in the LCD can be grouped in different meta-categories, according to their grammar. This classification can be thought of in terms of Aristotle's famous *square of opposition*. We can distinguish between: i) Universal affirmative statements, i.e. the (positive) features that apply to the whole class under consideration. As an example, we may consider the statements "a Fish is an Animal", or "a Kitchen Utensil is a Tool". Those statements can be treated as simple class inclusion, and in First Order Logic would correspond to universal quantification ("all fish are animal", etc). ii) Existential statements, which apply only to some instances of the class at hand: e.g. *Insect* "can bite", or *Tool* "can be automatised". In First Order Logic they would correspond to existential quantification: some insect bites, some tool is automatized, etc. iii) Universal negations, which apply again to the whole category, but which express a negated statement, like *Fish* "does not live on land" or *Insect* "does not live long". iv) Existential negation, of the kind "some A are not B", and which apply only to a subclass of the concept under consideration: e.g. a *Vegetable* "is not always green".

Table summarises the distribution of the features in the different meta-categories. As it can be seen in the table, most of the features enter the meta-category "Universal affirmative", while the negated statements (Universal negative and Existential negative) are very few.

Type of Statement	Frequency
Universal affirmative	$\approx 82\%$
Existential affirmative	$\approx 16\%$
Universal negative	$\approx 1\%$
Existential negative	$\approx 0,5\%$

Table 1: Features classification

Looking beyond the syntactic surface, however, within the Universal affirmative statements, only a few (less than 10%) are true, clear universal statements, which are valid for the whole category. Many other features (see e.g. the column 'Syntax' in Table 1) look like universal statements, but presuppose the possibility of exceptions: e.g. "Birds eat worms" is used as a *default* statement about birds, but it is possible to think of counterexamples, since not all birds are carnivores. When translating the features into axioms, it is desirable to distinguish between the axioms which require a classical, non-defeasible, use of the SubClassOf relation (e.g. *Bird SubClassOf Animal*), and axioms which do require a defeasible semantics (e.g. *Bird SubClassOf eats some Worm*). This distinction is registered at the level of annotation, and can be guided in different ways. In part, it is guided by the information in the database: we can in fact use the features' production frequency and their average judgments to take some decisions. The features which are generated often and which get a high average rating are more likely to be valid for the whole category. However, this strategy alone does not always guarantee satisfactory results. The feature "has feathers", for instance, was produced for the concept *Bird* by all the subjects involved in the experiment, and got the highest rating. However, it would be reasonable to make it a defeasible axiom, since e.g. many pullets do not have feathers.

A Game of Disambiguation

Foundational or upper ontologies (FO) formalise the meaning of very general terms, such as object, event, property, quality, relation, process, etc. (Borgo, Galton, and Kutz, 2022). They provide the top-level categories that are in principle common to many domains of application, and are implicitly at work in common sense. There are a number of different such ontologies which reflect different philosophical views on reality, ranging from a realistic stance endorsed by BFO (Arp, Smith, and Spear, 2015) to a cognitivistic perspective enabled by DOLCE (Masolo et al., 2002). While we do not take a position here about which is the right FO to analyse commonsense concepts, we stress that embracing the perspective of a selected FO has important consequences on the formal rendering of the commonsense expressions. For the sake of this discussion and for highlighting the use of FOs in general in representing commonsense concepts, we exemplify how a number of features in the Leuven concept database can be construed by means of a foundational analysis. Despite the disambiguation choices we propose here, some of the features in the database were still too idiosyncratic to fit modelling and logical rendering strategies, and were therefore manually discarded.

We here combine a two-level approach. Firstly, we identify a candidate categorical statement elicited from the LCD (e.g. All As are Bs). Secondly, we use FOs and their distinctions to help in identifying the intended meaning of classes A and B and in understanding the relevant representational choices. Although this section is descriptive in nature, it provides the basic rules of a **game of disambiguation** governed by foundational choices and representational modes. We therefore organise the discussion along **7 basic modes of disambiguation**:

Mode 1: Rigidity and anti-rigidity Two important general properties of classes are *rigidity* and *anti-rigidity*, cf. Guarino and Welty (2004). A *rigid* class is such that every instance of that class is *necessarily* an instance of that class. For example, in Figure 1, the feature *Animal* can be intended as a rigid class: a bird is an animal and, throughout its life, cannot cease to be an animal. An *anti-rigid* class is a class such that its instances eventually cease to be instances of that class. For example, in Figure 1 we have the feature *Migratory*. This class can be interpreted as an anti-rigid class, a *phase* of the life of the birds which has a beginning and an end. So when we represent the Leuven entries by means of axioms such as "all birds are animals" and "some birds are migratory", we can refine the meanings of these two statements by categorising the features as rigid or anti-rigid[4]. The *rigidity* and *anti-rigidity* distinction then plays a role in the context of Universal vs Existential statements described in the previous section. We may have statements of the kind "all As are Bs", where B is a rigid property, which means all As are always Bs (e.g. all Bird are always Animal). But we may also have Existentials of the kind "some As are Bs", where B is a rigid property, which again means that "some As are *always* Bs" (e.g. "some Animals are Birds"). On the other hand, both Existentials and Universals can involve anti-rigid properties, e.g. "some Birds are Migratory" and "all Mammal (mothers) breastfeed its babies". These issues are, of course, closely related to the semantic complexities found in Aristotle's modal Syllogistic (Malink, 2013).

Mode 2: Mereology An important ontological aspect is mereology, the theory of part-whole relations. FOs usually contain an axiomatisation of mereology, which makes the meaning of parthood relations explicit. Although the parthood relation may be not overly manifested in the syntax of the description of a feature, a number of entries in the LCD contain statements about the parts of an entity. E.g. in Table 1, "have wings" clearly indicates a parthood relation. So the rendering of that statement may be an axiom that states that the class of birds is included in the class of things that are related, via the parthood relation, to the class of wings.[5] The parthood relation is widespread in almost all the domains of the LCD (with the exception of the concept "Profession"), and constitutes about 13% of the features. The general ontological notion of parthood is quite abstract and, in many cases, one of the specialised parthood relations has to be considered, e.g. functional parthood, necessary parthood, temporary parthood, etc.

Mode 3: Spatio-temporal relations / Image Schemas Many entries in the database (15%) specify possible places in which an entity can dwell, e.g. "lives in the wild", "found in trees" (Bird), but also "sold in clothes shop" (Clothes) or "is often found in action movies" (Weapon). For these cases, FOs usually reify spatial and temporal locations as particulars of the ontology and can express the fact that an entity is located at a certain place or time. For instance, we can introduce the class of entities "located in the wild". These classes can be analysed according to the rigidity vs anti-rigidity distinction that we introduced earlier to assess the strength of the constrain: it may be necessary for fish to live in the water, while only accidental (non-rigid) to live in a cage or in the wild. Particularly salient spatio-temporal relations, and also prevalent in the LCD, are image-schematic ones, such as *containment*, *support*, or *path-following*. The importance of image schemas in computational blending has been illustrated in detail by Hedblom, Kutz, and Neuhaus (2016).

Mode 4: Quality and quality spaces A number of entries refer to qualities—i.e. colours, shapes, sizes, weights, etc.—of the instances. Around 12% of the features could be understood as qualities. For these cases, FOs like DOLCE provide a quite sophisticated analysis of quality ascriptions, relying on Conceptual Spaces (Gärdenfors, 2000). This approach renders the ascription of colors by introducing a relation of *location* between a *quality* and its *quale*, e.g. between the colour of a fish and a particular value of it, such as "bluish grey", which belongs to a suitable conceptual space of colours.

Mode 5: Constitution Other entries in the LCD contain the expression "made of" which is usually associated to what ontologists term *constitution*. For instance, DOLCE has a well-developed theory of *constitution* that is capable of approaching classical philosophical puzzles involving the persistence conditions of a statue constituted by a lump of clay. In this context, all the claims about the constitution of objects pertain to the artifact domain: *Vehicles* are "made of Metal", *Clothes* are "made of Textile", etc. Around 5% of the features of the artefact domain fall in this category ($\approx 2,5\%$ when considering the whole set of features).

[4] Handling the distinction between rigidity and anti-rigidity for OWL formalisations is challenging due to the limited expressive power of DLs.

[5] Another technical issue is to specify that the bird has to have exactly two wings, but that this assumption is defeasible.

Mode 6: Action and ability There are entries (mostly in the domain of animals, which constitute around 16% of the features) that ascribe an ability to an *agentive object*. Agentivity is sometimes intended in a broad sense, including animals. In entries such as "swim", "flutters", "sings", the intended meaning is that the animal can perform certain types of actions, i.e. the proper ontological category to assign is that of *ability*. Other entries include the word *can* which is quite challenging due to at least two meanings of *can* which have been intensively studied in knowledge representation: *ability can* (e.g. "Birds can fly") and *opportunity can* (e.g. "Birds can be found in trees ", which does not indicate an ability of the bird).

Mode 7: Functionality and affordances Other general concepts may be found in applications of foundational ontologies, in mid-level ontologies, or in more specific domain ontologies. Deciding how general a concept is may be a matter of discussion, however we can indicate a few quite general concepts that have applications in representing Leuven entities. *Functionality* is a concept usually related to artifact ontologies. Functionalities are intended to represent the purpose or the use of an artifact. Functionality is suitable to represent features that are expressed by means of words like "use", such as in "it is used to prepare Food" (Kitchen Utensils), or "means", as in "is a faster means of transportation" (Vehicle). *Affordances* (Turvey, 1992) are related to functionality in that they suggest a possible use of the object involved. For instance, a "Kitchen Utensil can be used to cut things", i.e. affords cutting. Around 20% of the features entered this mode of disambiguation.

In Figure 1, the column 'Modes Of Disambiguation' shows an example of the application of this analysis to the concept **Bird**. Overall, around 15% of the features contained in the database escaped our classification according to these modes of disambiguation, most of which where from the domain of activities (Professions and Sports).

Exploiting Commonsense Knowledge: The example of concept hybridisation

We conclude the paper discussing an application of the resulting formalised commonsense knowledge in a concrete computational implementation for concept combination.

To this end, we here briefly discuss the approach of Righetti et al. (2021b), who recently proposed an algorithmic modelling of the process of concept combination, leveraging the refinement operators described in Confalonieri et al. (2020) (restricted to description logic $\mathcal{ALC}$), and the techniques of axiom weakening. The paper aims to imitate the process of making sense of 'impossible' hybrid combinations, i.e. combinations of clashing concepts into imaginary objects such as "a Vehicle that is a Fish". This is inspired by the empirical research in cognitive psychology identifying human heuristics for combining concepts that lack any obvious similarities (Hampton, 2017).

In the approach of Righetti et al. (2021b), concepts are represented as formal ontologies in Description Logic, and the combination process is, thus, rendered as an ontology integration task. Briefly, the authors propose a turn-based algorithm which is initiated with two input ontologies which need to be blended into a final ontology describing the combined concept. The authors tested the procedure on the combination of the concepts Fish and Vehicle to try to replicate one of the human combinations studied in the experiments of Hampton (2017)—namely, the Fish-Vehicle concept. In this case, the concepts of interest to be combined are not just dissimilar, but, when formalised in a logical language, jointly contain obvious and sometimes hard to resolve formal inconsistencies. When adding an axiom to the combination causes an inconsistency, the approach of axiom weakening is applied until a jointly consistent compromise is found. Intuitively, a general concept inclusion axiom of the form $C \sqsubseteq D$ can be weakened by either specialising the concept C to a smaller class, or by generalising the concept D to a larger class, w.r.t. a given reference ontology[6].

In order to replicate human concepts one needs some repository of commonsense knowledge as a source for the input ontologies. A straightforward, encyclopedic definition of the concept at hand, would hardly faithfully represent what people have in mind when blending concepts. Consider, for instance, the Fish definition from Wikipedia: "Fish are aquatic, craniate, gill-bearing animals that lack limbs with digits. Included in this definition are the living hagfish, lampreys, and cartilaginous and bony fish as well as various extinct related groups."[7]. This definition might quite easily be axiomatised in a formal ontology, and there exist different tools for automatic natural language to OWL translation (see the related work section above) which can be employed in the presence of such clear and precise definitions. However, when combining the concepts Fish and Vehicle, people consider much more mundane knowledge. For instance, when combining the two concepts, humans may notice that while a Fish eats Food (to stay alive), a Vehicle needs Fuel (to move) (Hampton, 2017). By exploiting a heuristic similar to the analogical mapping described by Fauconnier (1997), people would tend to *generalise* this information into "both Fish and Vehicle need some kind of Energy to move", thus creating an interesting analogy between Food and Fuel, which would further support the integration of the two concepts into the combination "Fish-Vehicle".

Fortunately, this is exactly the kind of commonsense information the Leuven concept database is permeated with, thus suggesting the concrete usefulness of a formalisation of the concepts contained in the database for practical AI applications. To give a further tangible example, we fed the implementation proposed by Righetti et al. (2021b)[8] with two concepts contained in the Leuven Database, namely Bird and Kitchen Utensil, previously formalised in OWL exploiting the disambiguation steps described in this paper. The concept of Kitchen Utensil-Bird was also one of the exam-

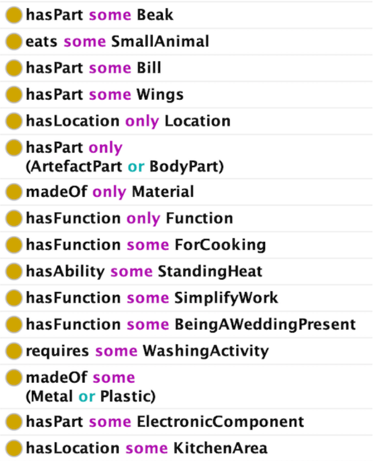
[6] We refer to the work of Righetti et al. (2021b) for the full details about the use of axiom weakening in this context.

[7] `https://en.wikipedia.org/wiki/Fish`.

[8] Available at `https://bitbucket.org/troquard/ontologyutils/src/master/`.

ples exploited by Hampton (2017) in his experiments on impossible combinations—see Figure 3.

Besides the features contained in the database, we included in the ontologies a few additional axioms, aiming at replicating some of the commonsense distinctions needed to reason about the concepts at issue but not explicitly mentioned by the subjects during the Leuven experiments because considered obvious or out of scope (as discussed above). We added, for example, the information that Animal and Tool are disjoint classes, or that if something is located in the kitchen it cannot (not normally, at least) be located on a tree, etc. An excerpt of one of the resulting ontologies for the concept KitchenUtensil-Bird is shown in Figure 2.

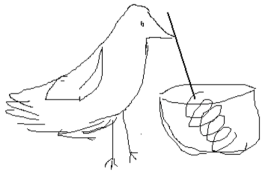


● hasPart some Beak
● eats some SmallAnimal
● hasPart some Bill
● hasPart some Wings
● hasLocation only Location
● hasPart only (ArtefactPart or BodyPart)
● madeOf only Material
● hasFunction only Function
● hasFunction some ForCooking
● hasAbility some StandingHeat
● hasFunction some SimplifyWork
● hasFunction some BeingAWeddingPresent
● requires some WashingActivity
● madeOf some (Metal or Plastic)
● hasPart some ElectronicComponent
● hasLocation some KitchenArea

Figure 2: A Bird which is also a Kitchen Utensil: an example of a blend exploiting the LCD information.

The procedure described by Righetti et al. (2021b) allows for a fine-grained selection of the combination strategies, by allowing the choice of a preference order over the axioms assigned to both agents/ontologies, as well as the distribution of turns. Also, different evaluation strategies are proposed to evaluate the outcome of the combination. Here, the example has just an illustrative purpose, and we set a random order over the axioms and an equal distribution of turns. However, the output of the procedure is surprisingly similar to the combination of the two concepts as observed in the experiments described by Hampton (2017), an example of which is presented in Figure 3.

As the output of our procedure, the *Whisking Woodpecker* has a beak and wings, thus showing body parts, but it also has artefact parts (the whisk), it is used for cooking and, being unhygienic, it also requires some washing.



"a woodpecker has been trained to whisk eggs using its powerful head movements (…) it would not need electrical power (good for camping trips) but would on the other hand be unhygienic."

Figure 3: The *Whisking Woodpecker* (Redrawn illustration as given by Hampton (2017), page 113): A woodpecker used to whisk as imagined by one of the participants of Hampton's experiments.

Discussion and Future Perspective

The analysis of the Leuven concept database has clearly shown the existence of a mismatch between the syntactic surface form and the content (or the *intended meaning*) of people's statements. On the one hand, we demonstrated this mismatch in the context of universal statements, where people often adopt a default reasoning strategy, and where the meaning they intend to convey is more likely close to an existential interpretation. On the other hand, many of the entries in the Leuven concept database lack a syntactic trigger that would help to identify their intended meaning, and ontological analysis is required to instead find a semantic trigger. So, for instance, the recognition of *mereological* assertions was mostly guided by our general language comprehension and competence for world understanding, but it was not manifested in the syntactical description of the features. Ontological analysis offers the means to explore systematically the possible meanings of commonsense feature ascriptions and, as a result, supports a more faithful formalisation into OWL.

We have also highlighted the fruitful connection between commonsense knowledge extraction and computational conceptual blending. Specifically, we have illustrated the practicality of this connection in a concrete computational workflow and use-case. Although illustrative in purpose, the example showed the effectiveness of the proposed methodology in replicating human conceptual combination as observed in the context of experimental psychology. In this context, the modes of disambiguation, as well as the syntactic analysis described above, could be exploited further, as a way to steer and guide the blending process. One may, for instance, integrate the dialogue implementation discussed above (Righetti et al., 2021b) to take into account such information, e.g. through the application of appropriate preference orders over the axioms of the two agents. This way one may prefer axioms involving universal, rigid statements, thus preserving them during the combination, and instead prefer the weakening of default statements, thus simulating a process similar to defeasible inference. Therefore, we plan to develop this further in future work towards more fine-grained evaluation metrics for blends and their creativity, in which essence fights serendipity.

Author Contributions

G. R. was the author mainly responsible for analysing the data. Aside from this, all authors equally contributed to the research presented in this paper.

Acknowledgments

This research is partially supported by Italian National Research Project PRIN2020 2020SSKZ7R.

References

Antoniou, G., and van Harmelen, F. 2009. Web ontology language: OWL. In Staab, S., and Studer, R., eds., *Handbook on Ontologies*, International Handbooks on Information Systems. Springer. 91–110.

Arp, R.; Smith, B.; and Spear, A. D. 2015. *Building Ontologies with Basic Formal Ontology*. Mit Press.

Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; and Ives, Z. G. 2007. Dbpedia: A nucleus for a web of open data. In Aberer, K., and al., eds., *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007*, volume 4825 of *Lecture Notes in Computer Science*, 722–735. Springer.

Bateman, J.; Beetz, M.; Beßler, D.; Bozcuoğlu, A. K.; and Pomarlan, M. 2018. Heterogeneous Ontologies and Hybrid Reasoning for Service Robotics: The EASE Framework. In Ollero, A.; Sanfeliu, A.; Montano, L.; Lau, N.; and Cardeira, C., eds., *ROBOT 2017: Third Iberian Robotics Conference*, 417–428. Cham: Springer International Publishing.

Boden, M. A. 1998. Creativity and artificial intelligence. *Artificial intelligence* 103(1-2):347–356.

Borgo, S.; Galton, A.; and Kutz, O. 2022. Foundational Ontologies in Action: Understanding foundational ontology through examples. *Applied Ontology* 17(1).

Britz, K., and Varzinczak, I. J. 2017. Context-based defeasible subsumption for dsroiq. In Gordon, A. S.; Miller, R.; and Turán, G., eds., *Proc. of the 13th International Symposium on Commonsense Reasoning, COMMONSENSE 2017*, volume 2052 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Buchanan, E. M.; Valentine, K. D.; and Maxwell, N. P. 2019. English semantic feature production norms: An extended database of 4436 concepts. *Behavior Research Methods* 51(4):1849–1863.

Casini, G., and Straccia, U. 2010. Rational closure for defeasible description logics. In Janhunen, T., and Niemelä, I., eds., *Logics in Artificial Intelligence - 12th European Conference, JELIA 2010*, volume 6341 of *Lecture Notes in Computer Science*, 77–90. Springer.

Confalonieri, R.; Galliani, P.; Kutz, O.; Porello, D.; Righetti, G.; and Troquard, N. 2020. Towards even more irresistible axiom weakening. In Borgwardt, S., and Meyer, T., eds., *Proc. of the 33rd International Workshop on Description Logics (DL 2020)*, volume 2663 of *CEUR Workshop Proceedings*. CEUR-WS.org.

De Deyne, S.; Verheyen, S.; Ameel, E.; Vanpaemel, W.; Dry, M. J.; Voorspoels, W.; and Storms, G. 2008. Exemplar by feature applicability matrices and other dutch normative data for semantic concepts. *Behavior Research Methods* 40(4):1030–1048.

Draicchio, F.; Gangemi, A.; Presutti, V.; and Nuzzolese, A. G. 2013. FRED: from natural language text to RDF and OWL in one click. In Cimiano, P.; Fernández, M.; López, V.; Schlobach, S.; and Völker, J., eds., *The Semantic Web: ESWC 2013 Satellite Events, Revised Selected Papers*, volume 7955 of *Lecture Notes in Computer Science*, 263–267. Springer.

Emani, C. K.; Silva, C. F. D.; Fiés, B.; and Ghodous, P. 2019. NALDO: from natural language definitions to OWL expressions. *Data Knowl. Eng.* 122:130–141.

Eppe, M.; Maclean, E.; Confalonieri, R.; Kutz, O.; Schorlemmer, M.; Plaza, E.; and Kühnberger, K.-U. 2018. A computational framework for conceptual blending. *Artificial Intelligence* 256:105–129.

Fauconnier, G., and Turner, M. 1998. Conceptual integration networks. *Cognitive science* 22(2):133–187.

Fauconnier, G. 1997. *Mappings in thought and language*. Cambridge University Press.

Fellbaum, C., and Hicks, A. 2019. When WordNet Met Ontology. In Borgo, S.; Ferrario, R.; Masolo, C.; and Vieu, L., eds., *Ontology Makes Sense - Essays in honor of Nicola Guarino*, volume 316 of *Frontiers in Artificial Intelligence and Applications*, 136–151. IOS Press.

Fellbaum, C. 2005. WordNet and wordnets. In Brown, K., ed., *Encyclopedia of Language and Linguistics*, 665–670. Oxford: Elsevier.

Gangemi, A.; Guarino, N.; Masolo, C.; and Oltramari, A. 2003. Sweetening WordNet with DOLCE. *AI Magazine* 24(3):13.

Gangemi, A.; Nuzzolese, A. G.; Presutti, V.; Draicchio, F.; Musetti, A.; and Ciancarini, P. 2012. Automatic Typing of DBpedia Entities. In Cudré-Mauroux, P., and al., eds., *The Semantic Web – ISWC 2012*, 65–81. Berlin, Heidelberg: Springer.

Gärdenfors, P. 2000. *Conceptual Spaces: The Geometry of Thought*. MIT Press.

Gonçalves, J.; Martins, P.; and Cardoso, A. 2017. Blend City, BlendVille. In *ICCC-17*, 112–119.

Grice, H. P. 12 Dec. 1975. *Logic and Conversation*. Leiden, The Netherlands: Brill. 41 – 58.

Guarino, N., and Welty, C. A. 2004. An Overview of OntoClean. In *Handbook on Ontologies*. Springer. 151–171.

Hampton, J. A. 2017. Compositionality and concepts. In Hampton, J. A., and Winter, Y., eds., *Compositionality and Concepts in Linguistics and Psychology*. Cham: Springer International Publishing. 95–121.

Hedblom, M. M.; Kutz, O.; and Neuhaus, F. 2016. Image schemas in computational conceptual blending. *Cognitive Systems Research* 39:42–57.

Hedblom, M. M.; Righetti, G.; and Kutz, O. 2021. Deciphering The Cookie Monster: A Case Study in Impossible Combinations. In de Silva Garza, A. G.; Veale, T.; Aguilar, W.; and Pérez y Pérez, R., eds., *Proceedings of the Twelfth International Conference on Computational Creativity*, 222–225. Association for Computational Creativity (ACC).

Hofstadter, D. R. 2001. *Epilogue: Analogy as the core of cognition*. Cambridge: MA: MIT Press. 499 – 538.

Krieg-Brückner, B.; Autexier, S.; Rink, M.; and Ghomsi Nokam, S. 2015. Formal Modelling for Cooking Assistance. In De Nicola, R., and Hennicker, R., eds., *Software, Services, and Systems*, volume 8950 of *Lecture Notes in Computer Science*. Springer International Publishing. 355–376.

Lenat, D. B. 1995. CYC: A Large-Scale Investment in Knowledge Infrastructure. *Commun. ACM* 38(11):33–38.

Malink, M. 2013. *Aristotle's Modal Syllogistic*. Harvard University Press.

Masolo, C.; Borgo, S.; Gangemi, A.; Guarino, N.; Oltramari, A.; and Schneider, L. 2002. WonderWeb deliverable D17. The wonderWeb Library of Foundational Ontologies and the DOLCE ontology.

McCarthy, J. 1959. Programs with Common Sense. In *Proceedings of the Teddington Conference on the Mechanization of Thought Processes*, 75–91. London: Her Majesty's Stationary Office.

McGuinness, D. L., and van Harmelen, F. 2004. OWL Web Ontology Language Overview. Technical report, W3C Recommendation.

Neuhaus, F.; Kutz, O.; Codescu, M.; and Mossakowski, T. 2014. Fabricating monsters is hard: towards the automation of conceptual blending. In *Proc. of the 3rd Int. Workshop on Computational Creativity, Concept Invention, and General Intelligence (C3GI@ECAI-14)*.

Nguyen, T.; Razniewski, S.; and Weikum, G. 2021. Advanced semantics for commonsense knowledge extraction. In Leskovec, J.; Grobelnik, M.; Najork, M.; Tang, J.; and Zia, L., eds., *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, 2636–2647. ACM / IW3C2.

Ontanón, S., and Plaza, E. 2010. Amalgams: A formal approach for combining multiple case solutions. In *International Conference on Case-Based Reasoning*, 257–271. Springer.

Porello, D.; Kutz, O.; Righetti, G.; Troquard, N.; Galliani, P.; and Masolo, C. 2019. A toothful of concepts: Towards a theory of weighted concept combination. In Simkus, M., and Weddell, G. E., eds., *Proc. of the 32nd International Workshop on Description Logics*, volume 2373 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Quine, W. V. O. 1960. *Word & Object*. MIT Press.

Righetti, G.; Porello, D.; Kutz, O.; Troquard, N.; and Masolo, C. 2019. Pink panthers and toothless tigers: three problems in classification. In Cangelosi, A., and Lieto, A., eds., *Proc. of the 7th International Workshop on Artificial Intelligence and Cognition*, volume 2483 of *CEUR Workshop Proceedings*, 39–53. CEUR-WS.org.

Righetti, G.; Masolo, C.; Troquard, N.; Kutz, O.; and Porello, D. 2021a. Concept combination in weighted logic. In *Proceedings of the Joint Ontology Workshops 2021, Episode VII*, volume 2969 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Righetti, G.; Porello, D.; Troquard, N.; Kutz, O.; Hedblom, M. M.; and Galliani, P. 2021b. Asymmetric Hybrids: Dialogues for Computational Concept Combination. In *Formal Ontology in Information Systems (FOIS 2021)*, volume 344 of *Frontiers in Artificial Intelligence and Applications*, 81–96. IOS Press. FOIS Best Paper Award.

Rosch, E. H. 1973. On the internal structure of perceptual and semantic categories. In Moore, T. E., ed., *Cognitive Development and Acquisition of Language*. San Diego: Academic Press. 111–144.

Ruts, W.; De Deyne, S.; Ameel, E.; Vanpaemel, W.; Verbeemen, T.; and Storms, G. 2004. Dutch norm data for 13 semantic categories and 338 exemplars. *Behavior Research Methods, Instruments, & Computers* 36(3):506–515.

Schmidt, D.; Pease, A.; Trojahn, C.; and Vieira, R. 2019. Aligning conference ontologies with SUMO: A report on manual alignment via wordnet. In Barton, A.; Seppälä, S.; and Porello, D., eds., *Proc. of the Joint Ontology Workshops 2019, Episode V*, volume 2518 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Silva, V. S.; Freitas, A.; and Handschuh, S. 2016. Word Tagging with Foundational Ontology Classes: Extending the WordNet-DOLCE Mapping to Verbs. In *20th International Conference on Knowledge Engineering and Knowledge Management - Volume 10024*, EKAW 2016, 593–605. Berlin, Heidelberg: Springer-Verlag.

Speer, R.; Chin, J.; and Havasi, C. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In Singh, S. P., and Markovitch, S., eds., *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, 4444–4451. AAAI Press.

Storms, G.; De Boeck, P.; and Ruts, W. 2000. Prototype and exemplar-based information in natural language categories. *Journal of Memory and Language* 42(1):51–73.

Turvey, M. T. 1992. Affordances and prospective control: An outline of the ontology. *Ecological psychology* 4(3):173–187.

Veale, T. 2019. From Conceptual Mash-ups to Badass Blends: A Robust Computational Model of Conceptual Blending. In *Computational Creativity*. Springer. 71–89.

Vinson, D. P., and Vigliocco, G. 2008. Semantic feature production norms for a large set of objects and events. *Behavior Research Methods* 40(1):183–190.

Völker, J.; Hitzler, P.; and Cimiano, P. 2007. Acquisition of OWL DL axioms from lexical resources. In Franconi, E.; Kifer, M.; and May, W., eds., *In proc. ESWC 2007*, volume 4519 of *Lecture Notes in Computer Science*, 670–685. Springer.

Yeh, W., and Barsalou, L. W. 2006. The situated nature of concepts. *The American Journal of Psychology* 119(3):349–384.

The Word-Weaving Clock: Time Constraints in Word Associations

Alessandro Valitutti
Phedes Lab
http://phedes.com
alessandro.valitutti@phedes.com

Abstract

A wise design of time constraints is crucial for the computational realization of several creative tasks. In this paper we focus on live generation of word associations unfolding semantic paths modulated by contextual clues. We present the *Word-Weaving Clock*, a lexical creative system performing the generation of word associations where both semantic and time constraints are taken in account. The system is meant to be used interactively as part of a live demonstration.

Introduction

Do time constraints promote creativity? And if it is true for human creativity, does this also apply to the computational one?

A good number of computational-creativity tasks, mostly inspired by Margaret Boden's seminal ideas (Boden, 1990), are defined as a search in a conceptual space. Unfortunately, search processes are time-consuming, especially in those cases that require access to a vast amount of common sense knowledge. This affects the feasibility of tasks in which a time-constrained performance is essential for the appreciation of creativity. Let's think, for example, of a musical jam session, freestyle rap, or poetry slam. There are forms of creative brainstorming and stand-up comedy where prompt responsiveness or interactivity are crucial aspects of the game.

In this paper, we focus on a specific task consisting of a live generation of word associations unfolding semantic paths modulated by contextual clues. We present the *Word-Weaving Clock*, a lexical creative system performing the generation of word associations where both semantic and time constraints are taken into account. Moreover, we report the conceptual elements and the resources employed in the design and implementation of the system. The task and the resources upon which it was built have been kept as simple as possible, to facilitate its performance replicability and its potential use as a component of more complex systems. Finally, we give examples of outputs suggesting how the system would behave as part of a live presentation.

Background

As reported by Haught and Johnson-Laird (2003), "constraints are at the heart of the creative process. They govern the generation of ideas". In particular, time constraints affect creative writing (Biskjaer et al., 2019). They should be tuned carefully because "creativity can be compromised by both scarcity and abundance of time" (Liikkanen et al., 2009; Baer and Oldham, 2006).

A major issue in applying time constraints is that search processes are necessarily time-consuming. There is no way to reduce arbitrarily the time required to perform knowledge discovery in a large dataset. For this reason, indexing is an essential aspect of the design of an information retrieval algorithm, as in the case of search engines (Zuze and Weideman, 2011). In the recent development of transfer learning, knowledge and language models can be effectively reused in a way to reduce machine-learning training time (Zhuang et al., 2021). Since both writing and ideation processes are based on the discovery and exploitation of links between concepts, computational creativity efforts has been put into lexical associations. Gross et al. (2012) focused the Remote Association Test, a task in which, given three words, the word semantically connecting all of them is required. They implemented it through co-occurrence frequencies of word pairs in a large textual corpus.

In our version of this task, we have two input words with different semantic roles. The *main word* generates a set of candidate words according to the relation modeling associations in the common-sense knowledge. The *clue word* – introduced either at the beginning or, interactively, at any point in the process – provides a semantic context and allows the system to make a selection from the candidate words. For instance, the main word *'eyes'* generates *'skin'* according to the *'body'* clue word, and *'night-sight'* according to the *'pleasure'* clue word. Most importantly, a *one-second time constraint* is imposed so that a new output word is generated every second. The overall aimed effect is providing creativity both in the interactive experience and in the generated path of word associations. This periodic recursion in the generation of word associations is meant to evoke the flow of consciousness emerging as blending of semantic pulses, as an inspiration from the notion of Damasio's core consciousness, which is seen as "created in pulses, each pulse triggered by each object that we interact with or that we recall" (Damasio, 1999).

Task Description

The task consists of the following elements:

- **Word Selection Step.** Firstly, the exploration of semantic relatedness, modeling associations in common-sense knowledge, allows the system to identify a set of candidate words related to the input word. Next, a further contextual semantic constraint (the *semantic slanting*) allows the system to select the *output word*.
- **One-Second Time Constraint.** It is prefixed as the one-second time interval within which the word selection step should be performed.
- **Iteration.** The word selection step is iterated so that the output word became the input word of the next step, thus generating an associative word path as the overall product of the interaction. To avoid repetitions, the words in the associative paths are removed from the candidate words in the next word selection step.

As an example of output, the word *'travel'* is associated, in the system, to the word path *"booking* → *hotel* → *casino-hotel* → *jack-in-the-box* → *..."*. With the clue *'happy'*, the path is *"go around* → *make up* → *know how* → *well-wishing* → *..."*, while with the clue *'unhappy'*, the path is *"go around* → *go slow* → *long-suffering* → *ill-being* → *..."*. Finally, the input word could be used as the clue word of itself, so reinforcing association closer to its semantic domain, e.g. producing *"booking* → *tour* → *visiting* → *shopping* → *..."*.

Implementation

1. Common-Sense Associations: Word Embeddings

One of the most effective ways to implement semantic relatedness of words (the so-called "word similarity") consists of the cosine distance between word pairs represented as vectors (Mikolov et al., 2013). To measure word similarities we employ word embedding provided by *Spacy*[1], an open-source software library in Python for advanced natural language processing (Hiippala, 2021; Jurafsky and Martin, 2000). In particular, we use word2vec model for word embedding Jatnika, Bijaksana, and Suryani (2019) trained it on a large-scale language model in English[2]. In the analysis of word similarity, the procedure filtered word pairs with similarity values greater or equal to 0.2.

2. Semantic Slanting: Clue Word

Once found a set of candidate words, all semantically related to the current word, the further selection is performed according to the semantic relatedness with the clue word slanting the search of associations toward a specific semantic domain or connotation. The clue word could represent an emotion or, more abstractly, a sentiment polarity (e.g., either positive or negative). In a possible live demonstration, the system is meant to insert or modify the clue word at runtime, in such a way as to modulate the associative path interactively.

3. Time Constraint: Indexing

We need to reduce the running time for each word selection step below a single second. The main bottleneck is measuring word similarity for all word pairs whose first term is the input word. Word similarity search is highly time-consuming. For instance, measuring word similarity with Spacy takes an average time of 20 milliseconds (with a CPU clock speed of 3.2 GHz). So measuring similarity with 3000 words is sufficient to exceed the one-second time constraint. Oxford Dictionary has 273,000 headwords, 71,476 of them being in current use[3]. So if we want to compare an input word with all headwords, the running time would be more than one hour and a half. Therefore, it is clear that only smartly-designed indexing allows the system to satisfy the one-second time constraint.

The following points summarize the choices that allowed us to satisfy the time constraint:

1. We collected from the Web about 27600 English nouns. We used WordNet as a tool for selecting nouns (Miller, 1995).
2. Next, we randomized the list of nouns and partitioned it into a number of sublists of the size of about 3000 items, then we calculated mutual similarities inside each sublist. In this way, we were able to develop the full system (with the time constraint) with an increasing efficiency according to the current state of indexing.
3. Finally, we calculated similarities between different sets, adding a subset a time until all mutual similarities were indexed.

In general, the number of similarity measurements N_{sm} for a word set of size n is:

$$N_{sm} = \frac{(n-1)\cdot n}{2} - 1$$

Therefore, the previewed overall running time for the 12000 nouns plus three subsets of 3000 items takes about three weeks of computation. The full set of 27000 nouns would need 84 days of computation.

As part of this research contribution, we provided the indexed resource in a text file[4]. To implement fast indexing and retrieval, we created a database using sqlite3[5] – Python interface to SQLite[6] database engine library. In the current version of the *Word-Weaving Clock*, most of the contribution to running time is mostly consisting in searching the table rows with similarity values, having the input word as either the first or the second element of the word pair.

[1] https://spacy.io

[2] spacy.io/models/en#en_core_web_lg

[3] https://en.wikipedia.org/wiki/List_of_dictionaries_by_number_of_words – retrieved May 28, 2022.

[4] https://www.kaggle.com/datasets/alessandrovalitutti/noun-similarity-pairs

[5] https://docs.python.org/3/library/sqlite3.html

[6] https://www.sqlite.org/index.html

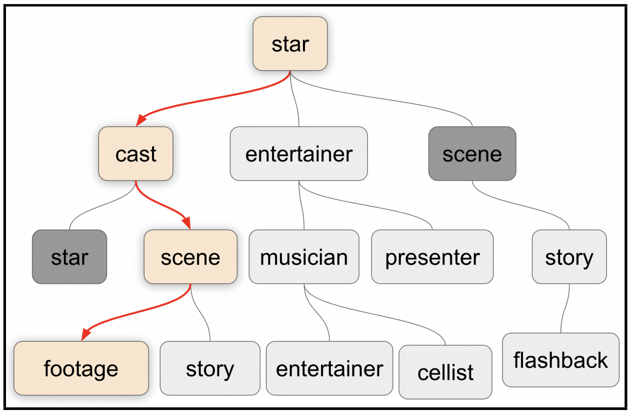


Figure 1: Example of similarity word tree as the composition of indexed word-similarity associations.

Examples of Outputs

Figure 1 shows an example of indexed common-sense word associations (without semantic slanting). Starting from the root word *'star'*, the procedure selects the node containing the word most similar to the word in the current parent node. Dark grey nodes contain words already included in the associative path, thus removed from the future selections.

Table 1 compares different possible associative paths generated from the word *'star'* and corresponding to different clue words (from the second to the last column) or without semantic slanting (first column). Clue words with opposite polarity (e.g., *'success'* vs. *'failures'*) modulate the generation of paths with positive and negative sentiment, respectively (columns two and three). In particular, emotion words with opposite polarity (e.g., *'joy'* vs. *'disgust'*) allow the procedure to generate paths characterized by affective valence (columns four and five). Moreover, using a domain word as a clue word (such as *'physics'* or *'movie'* induces a reinforcement in the generation of words in that domain (see columns six and seven). Finally, comparing columns one (no clue word) and seven (*'movie'* as clue word), we can see that semantic slanting tends to keep word associations in the same domain, although without semantic slanting the path can include more domains with higher semantic cohesion (e.g. *cinema* and *music* in column 1).

We emphasize that the reported examples are generated with a preliminary version of the similarity database. Next versions will access associations with higher values of word similarity and corresponding quality in the semantic cohesion.

Conclusions

The *Word-Weaving Clock* has been conceived primarily to stimulate interest in the study of timing in computational-creativity tasks. Time constraints are meant to be taken into account not only to improve user experience but also the design process itself since it challenges the designer to perform a wise balance between offline indexing and runtime running. Although the main intended creative value is in the interactivity experience that comes from a live demonstration of the system, the produced associative path can be considered as an artifact exhibiting creative value per se. A testbed version of the system could be used for the offline exploration of alternate paths according to different semantic and temporal parameters, for identifying values and ranges useful to improve the interactive version. The provided dataset of similarity values on English nouns is a potential handful resource for allowing researchers to further explore common-sense associations without the need for indexing.

As a possible application, time beat synchronized associative paths could be used as a backbone for the real-time selection of tweets or poetic lines. Semantic slanting could be performed according to more complex semantic patterns modeling personality traits. For example, the alternation of positive and negative slanting words could be used to mimic emotional instability.

Our next step in the development of the proposed system is to explore processing threads, to make it capable of performing both information indexing and retrieval concurrently. In some contexts, it would be interesting providing the system with full autonomy in analyzing a set of texts and building its model of word embedding, according to which the word associations will be discovered. Finally, we intend to explore different scenarios and user interfaces to make the system useful as a testbed for the offline exploration of associative paths.

Author Contributions

A.V. ideated and wrote the paper alone.

Acknowledgements

This research has received no external funding.

References

Baer, M., and Oldham, G. R. 2006. The curvilinear relation between experienced creative time pressure and creativity: Moderating effects of openness to experience and support for creativity. *Journal of Applied Psychology* 91(4):963–970.

Biskjaer, M. M.; Frich, J.; Vermeulen, L. M.; Remy, C.; and Dalsgaard, P. 2019. How time constraints in a creativity support tool affect the creative writing experience. In *Proceedings of the 31st European Conference on Cognitive Ergonomics (ECCE 2019)*, 100–107.

Boden, M. A. 1990. *The Creative Mind*. London: Abacus.

Damasio, A. 1999. *The Feeling of What Happens: Body and Emotion in the Making of Consciousness*. New York: Harcourt Brace.

Gross, O.; Toivonen, H.; Toivanen, J. M.; and Valitutti, A. 2012. Lexical creativity from word associations. In *International Conference on Knowledge, Information and Creativity Support Systems (KICSS)*, 35–42.

Haught, C., and Johnson-Laird, P. N. 2003. Creativity and constraints: The production of novel sentences. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 25, 528–532.

	success	**failure**	**joy**	**disgust**	**physics**	**movie**
cast	role	role	glamour	scene	role	flick
scene	success	concern	thrill	horror	general	soundtrack
footage	progress	failure	enjoyment	trepidation	assumption	trailer
trailer	confidence	fault	happiness	distaste	terms	scene
soundtrack	dedication	integrity	elation	exasperation	language	horror
intro	attention	disregard	emotion	contempt	arithmetic	story
interlude	influence	unwillingness	tears	stupidity	quantum	footage
accompaniment	concern	inefficiency	solace	rudeness	geometry	watching
choral	failure	reluctance	tranquility	unwillingness	singularity	streaming
oratorio	integrity	inconsistency	joyfulness	uneasiness	neutrino	sex
castrato	sustainability	assumption	faithfulness	elation	climatology	blonde
mezzo	partnership	burden	dedication	emotion	oceanography	girlfriend
pianissimo	milestone	severity	confidence	fascination	anthropology	thought
cadenza	challenge	condition	vitality	anticipation	lecturer	sort
hapsichord	make	necessity	imagination	anxiety	grad	make

Table 1: Example of associative word paths generated from from the input word *'star'* and modulated by different clue words.

Hiippala, T. 2021. Applied Language Technology: NLP for the Humanities. In *Proceedings of the Fifth Workshop on Teaching NLP*.

Jatnika, D.; Bijaksana, M. A.; and Suryani, A. A. 2019. Word2Vec Model Analysis for Semantic Similarities in English Words. *Procedia Computer Science* 157:160–167.

Jurafsky, D., and Martin, H. J. 2000. *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*. Upper Saddle River, N.J.: Prentice Hall.

Liikkanen, L. A.; Björklund, T. A.; Hämäläinen, M. M.; and Koskinen, M. P. 2009. Time constraints in design idea generation. In *Proceedings of the International Conference on Engineering Design (ICED'09)*.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS'13)*, volume 2, 3111–3119.

Miller, G. 1995. Wordnet: a lexical database for english. *Communications of the ACM* 38(11):39–41.

Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; and He, Q. 2021. A comprehensive survey on transfer learning. *Proceedings of the IEEE* 109(1):43–76.

Zuze, H., and Weideman, M. 2011. A comparative analysis of search engine indexing time. In *Proceedings of the 13th Annual Conference on World Wide Web Applications*.

Generation and Evaluation of Creative Images from Limited Data: A Class-to-Class VAE Approach

Xiaomeng Ye, Ziwei Zhao, David Leake and David Crandall
Luddy School of Informatics, Computing, and Engineering
Indiana University
Bloomington IN 47408, USA
{xiaye,ziwei}@iu.edu, {leake, djcran}@indiana.edu

Abstract

Generating novel items with desired characteristics requires creativity. One method to achieve this is through creative transformations. Deep learning network methods provide an interesting potential substrate for this task. This paper presents a method for network-based generation of novel images by applying variational autoencoders (VAEs) to learn features, which are then perturbed based on a class-to-class (C2C) method for learning of inter-class similarity and difference information, enabling generating creative samples. Our method learns the pattern between classes, applies this pattern to samples of a source class, and generates new samples of a target class. This study also proposes a general approach to evaluating the creativity of sample generators for classification domains, by evaluating the samples generated by the generator trained in a one-shot setting. The evaluation approach requires only classification labels but not human assessments of creativity. An experiment in two image domains supports that the samples generated by our method satisfy two of Boden's creativity criteria: being valuable (falling into desired categories) and novel (samples show high variance).

Introduction

Advances in machine learning have yielded many successful deep generative models (Pan et al. 2019). Such models generate samples conforming to the distribution of the training data, leading to samples that are "authentic" in the sense of substantially sharing the properties of real examples. A surge of research in computational creativity is applying generative deep learning methods while inducing novelty—and even surprise—for the sake of creativity, as described in the surveys of Franceschelli and Musolesi (2021) and Broad et al. (2021). For example, the creative adversarial network proposed by Elgammal et al. (2017) is a generative adversarial network (GAN) that generates artwork that is realistic but also deviates from style norms. Similar work has induced creativity in GANs by introducing additional goals (loss functions) beyond the original adversarial loss (e.g. StyleGAN (Karras, Laine, and Aila 2018) and (Sbai et al. 2018)). Following StyleGan, Nobari, Rashad, and Ahmed (2021) proposed a systematical method to modify GANs to automatically generate novel designs without human intervention. Generally, variational auto encoder (VAE) methods are less suited for creativity tasks because their reconstruction loss aims to mimic the data distribution within a learned latent space and it is difficult to reflect other goals in the corresponding loss function. However, these latent spaces can be manipulated to induce creative results (e.g. MusicVAE (Roberts et al. 2018b) and sketchRNN (Ha and Eck 2017)).

Characterizing the creativity of AI systems requires criteria for assessing the creativity of a process or of a system's results within a task context. Developing such criteria is nontrivial and has received considerable attention (Wiggins 2021). Boden (1991) provides three criteria for assesssing the creativity of outputs of a process: value, novelty, and surprise. Many researchers have continued this school of thought, refining and expanding on these criteria (Wiggins 2006; Draper 2010).

This paper addresses creativity as it applies to generating new samples for a target class when training samples are limited. We consider a sample (generated or not) to be *valuable* if it fits in the target class, and *novel* if it is different from the observed samples of the target class. According to Boden, surprise can happen when the sample is unexpected (which requires a prior expectation entity). We do not consider surprise when evaluating our model.

This paper makes two contributions. First, we present an algorithm for generating creative samples in a classification domain. The algorithm uses a method we call a *class-to-class variational autoencoder* (C2C-VAE), which learns a latent space of the difference patterns between samples of all classes. The C2C-VAE then samples new differences from this latent space, and applies the difference to existing samples in the original conceptual space to generate new samples. Second, we address the general question of how to evaluate the creativity of sample generators for classification in a one-shot setting. We propose an approach which we call GOF/TOM—"Generated On Few, Tested On Many." A generator is trained in a zero, one, or few-shot setting where samples of a target class are trimmed from the training set. The generator is then used to generate new samples of the target class. Meanwhile, an oracle is trained on the *untrimmed* dataset to evaluate the generated samples. Because the generator has limited examples of the target class, its ability to generate satisfactory unseen samples of the target class can be used as measure for its creativity. This is used to evaluate the C2C-VAE.

We begin by discussing related techniques for generating and measuring computational creativity. We then present our C2C-VAE approach for generating creative samples, and introduce our GOF/TOM approach for evaluating creativity. Finally, we evaluate C2C-VAE on two data sets, MNIST (LeCun and Cortes 2010) and Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017), using the GOF/TOM approach. In the two data sets, C2C-VAE successfully generates samples that are valuable and novel with respect to its training data, making a case for the potential of C2C-VAE as a creative approach. We examine the limitations of C2C-VAE and propose methods for addressing them in future work.

Background

Active Divergence with Generative Deep Learning

In her seminal work, Boden (1991) identifies three forms of creativity: combinatorial, exploratory and transformational. Combinatorial creativity generates new ideas by combining old ones. Exploratory and transformational creativity both involve a conceptual space, where the former explores the conceptual space while the later alters it, potentially causing a paradigm shift (Wiggins 2006; Franceschelli and Musolesi 2021).

Franceschelli and Musolesi (Franceschelli and Musolesi 2021) consider VAEs and GANs to perform exploratory creativity, as they both sample from a conceptual space. GANs can be also be transformational. As an example, in CANs (Creative Adversarial Networks), the discriminator determines both whether a sample image is art or not and its artistic style, while the generator tries to generate art and also generates deviations from original style norms. GANs can even be combinatorial. For example, StyleGAN can achieve style mixing by combinig the latent codes of two samples at multiple different levels of detail.

A CycleGAN is a image-to-image translation technique that can translate an image of one class into an image of another, e.g. modifying the image of a horse h into that of a zebra z using a translation function Z (or conversely from a zebra into a horse, using a function H). A CycleGAN is trained with two loss functions: 1) An adversarial loss trains the generators Z and H to generate quality images (so that a horse h can be translated into a realistic zebra $Z(h)$); 2) A cycle-consistency loss ensures the transition can go back-and-forth (so that the horse-translated zerba $Z(h)$ can be translated back to a horse $H(Z(h))$ similar to the original horse h). The artist Helena Sarin uses CycleGAN to generate creativity-related artwork (NVIDIA 2021).

Our proposed model is based on variational autoencoders (VAEs) (Kingma and Welling 2013). A VAE is comprised of an encoder and a decoder, both implemented as neural networks. The encoder takes samples as inputs and compresses them into a Gaussian distribution of lower-dimension embedding vectors in a latent space. The decoder takes an embedding vector and recovers the original input sample. Regularity—the property of similar samples having similar representations—is encouraged in the latent space because a sample is encoded as a distribution of embeddings, instead of a single embedding as in autoencoders (the forerunners of VAEs).

Features extracted by VAE can be manipulated by perturbation and even vector arithmetic for creative results. For example, MusicVAE (Roberts et al. 2018b) has the ability to "adjust the number of notes in a melody by adding/subtracting a note density vector to/from the latent code" (Roberts et al. 2018a). Similarly, sketchRNN can "subtract the latent vector of an encoded pig head from the latent vector of a full pig, to arrive at a vector that represents a body. Adding this difference to the latent vector of a cat head results in a full cat (i.e. cat head + body = full cat)" (Ha and Eck 2017). The effects of such modification over VAE embeddings are not guaranteed and only partially understood. As noted by Ha and Eck (2017), such analogy is only possible when the embedding distribution is smooth and any interpolation between two embeddings is coherent. This study attempts to model the differences between pairs of embeddings extracted by VAE.

In the taxonomy of active divergence by Broad et al. (2021), this study proposes a method of chaining models. The method is a combination of a standard VAE with a secondary VAE (C2C-VAE) that explores the learned representation of feature differences.

Class-to-class Approach

Classification methods commonly consider the similarity of new instances to instances in a class. The Class-to-class (C2C) approach considers both similarity and difference. It assumes that there exist inter-class patterns between each pair of classes, and the samples from the two classes are consistently similar in some features and different in some other features. For example, zebras and horses have the similarity of both belonging to the Equidae family, and the difference that zebras have stripes while horses do not. The inter-class patterns, once learned, can be used to classify a query based on instances from another or multiple other classes (Ye 2018a; Ye et al. 2020; 2021).

We hypothesize that inter-class patterns can also be used in computational creativity. A system that learns inter-class difference patterns can intentionally apply the patterns to modify a sample. For example, knowing that zebras have stripes and horses do not, the system can modify a horse image by replacing its texture with black-and-white stripes and thus create a new zebra image.

The C2C approach is highly related to GAN methods. For example, CycleGAN is trained on unpaired image-to-image data from one class to another and can generate zebra images from horse images (Zhu et al. 2017). GAN methods are mostly end-to-end. For example, CycleGAN generates an output image from an input image, and the inter-class pattern is integrated into the procedure of the model and is applied automatically in the forward pass of the neural network. C2C methods work with the inter-class pattern directly. For example, the method to be presented in this study uses the feature differences between two samples as both inputs and expected outputs of an variational autoencoder. This difference provides more flexibility to introduce creativity. More specifically, our approach can choose an inter-class pattern as the modification and also choose a sample to apply this modification.

Measurement of Creativity

Franceschelli and Musolesi (2021) survey multiple creativity measures implemented via machine learning algorithms. Our method, GOF/TOM ("generated on few, tested on many"), fits within the formalization of the generate and test framework (Toivonen and Gross 2015), in which the system uses a generative function to generate samples and an evaluation function to evaluate the samples. The authors describe three works (Varshney et al. 2013; Norton, Heath, and Ventura 2010; Morris et al. 2012) that fit in this framework.

Varshney et al. (2013) proposes a system that generates creative recipes. The novelty of a recipe is evaluated based on Bayesian surprise, the difference between a prior probability distribution of recipe and a posterior probability distribution after a new recipe is observed. The value of a recipe is evaluated by a model predicting pleasantness of scent from its ingredients and flavor compounds in those ingredients.

Ritchie (2007) describes that creativity can come from an *inspiring* set, which is a set of usually highly valuable samples used to train or configure the generator. Gervás (2011) expands on this by splitting an inspiring set into a learning set, which informs the construction of the generator, and a *reference* set, which is used to evaluate the novelty of generated samples. Similarly, Morris et al. (2012) uses an inspiring set (crockpot recipes) to generate samples via a genetic algorithm and to evaluate the quality of generated sample by training a multilayer perceptron to predict user ratings from a sample.

Creativity Inspired Zero-Shot Learning

The goal of a zero-shot learning task for classification is to train on seen classes and then predict the class label of a sample from an unseen class (or samples from seen and unseen classes in generalized zero-shot learning). Elhoseiny and Elfeki (2019) implemented a creativity inspired zero-shot learning algorithm. In that work, both visual and semantic information are available for seen classes but only semantic descriptions are available for unseen classes. The authors introduce a creativity inspired zero-shot learning method which trains a discriminator to differentiate between real and fake images and also classifies an image into seen classes. It also trains a generator to generate realistic images based on texts describing seen classes and realistic yet hard-to-classify (high entropy over seen classes) images from "hallucinated" texts. This training goal drives the generator to explore the latent space of texts with two objectives: 1) Generate samples that are realistic; 2) Generate samples from hallucinated texts. These properties enable the generator to generate realistic images based on the descriptions of unseen classes.

Creative Sample Generation with a Class-to-class Variational Autoencoder

This paper proposes a class-to-class variational autoencoder (C2C-VAE) approach to generating creative samples in the context of classification—that is, generating creative samples falling within desired categories (e.g., generating images for creative versions of a given letter of the alphabet). For this task, the C2C-VAE approach learns the difference pattern between pairs of samples of two different classes. Four spaces are involved in this task: the original sample space $L1$, the feature space $L2$, the space of feature differences $L2'$, and the space of feature difference embeddings $L3$. As a precondition, C2C-VAE relies on a means to transition between $L1$ and $L2$, more specifically, to extract a feature $f(s)$ from a sample s and also to recover a sample s from feature $f(s)$.

For our testbed system (illustrated in Figure 1), we train a traditional VAE on the training data and use the encoder f to extract features and the decoder f' to recover samples. Given a pair of samples from different classes, $s_1 \in C_1$ and $s_2 \in C_2$, a VAE can extract their features $f(s_1)$ and $f(s_2)$ in the space $L2$. The feature difference $f_\triangle(s_1, s_2)$ in the space $L2'$ can be calculated as $f_\triangle(s_1, s_2) = f(s_1) - f(s_2)$, an element-wise subtraction between two feature vectors. The space $L2'$ can be thought of as the complement of the space $L2$, whence the name $L2'$.

C2C-VAE is based on the class-to-class assumption that the feature differences (in space $L2'$) from one class to another class follow a consistent pattern. This pattern can be represented as an embedding vector (or a distribution of embeddings) in another latent space $L3$. The C2C-VAE is itself another variational autoencoder with an encoder g that encodes a feature difference $f_\triangle$ in $L2'$ to an embedding $g(f_\triangle)$ in $L3$ and a decoder g' that decodes an embedding $g(f_\triangle)$ back to a feature difference $f_\triangle$. Note that both the VAE encoder f and the C2C-VAE encoder g are *variational encoders* that encode an input to a distribution of embeddings. This is to ensure regularity in the latent space $L2$ and $L3$. For simplicity, the encoder f (or g) can be thought of as extracting one feature (or a feature difference embedding) from an input.

Because there exist multiple pairs of classes and each pair $C_i - C_j$ has its own unique pattern, a C2C-VAE either learns only one pattern, reflecting a specific pair of classes $C_i - C_j$, or learns multiple patterns by conditioning its encoder and decoder with extra parameters indicating C_i and C_j. In our tests, we take the later approach. Therefore, the C2C-VAE presented here is actually a *conditional* variational autoencoder (Sohn, Lee, and Yan 2015).

Training a C2C-VAE

A C2C-VAE is trained using the following procedure:

- Train a traditional VAE with encoder f and decoder f'.
- Assemble training pairs: Randomly collect 10000 pairs of samples during every training epoch. For each sample pair s_i (of class C_i) and s_j (of class C_j), extract their features $f(s_i)$ and $f(s_j)$, calculate their feature difference $f_\triangle(s_i, s_j) = f(s_i) - f(s_j)$.

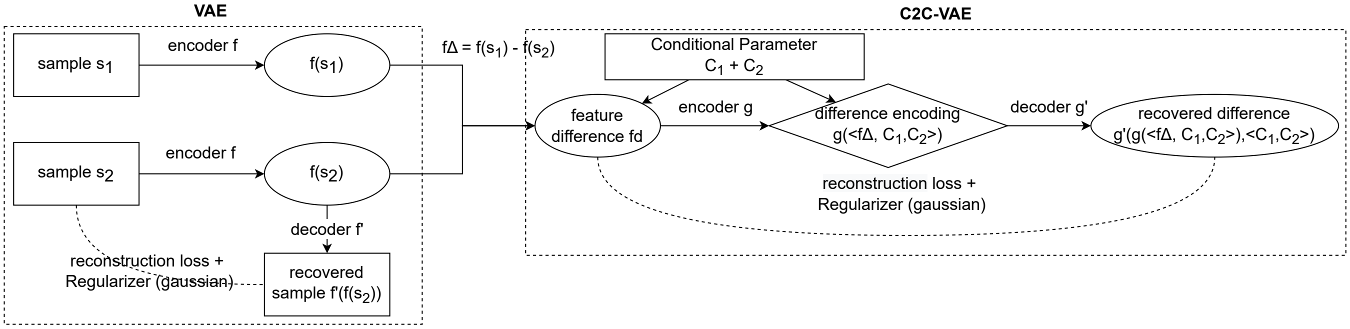


Figure 1: A VAE extracts (recovers) a feature from (to) a sample. A C2C-VAE extracts (recovers) an embedding from (to) a feature difference. $L1$ entities are marked with rectangles, $L2$ and $L2'$ entities with circles, and $L3$ entities with diamonds.

- Train C2C-VAE with the vector $< f_\triangle, C_i, C_j >$: The encoder g (conditioned on the class pairs) learns to encode the input to embedding $g(< f_\triangle, C_i, C_j >)$. The decoder g' (also conditioned on the class pairs) learns to decode the embedding back to $f'_\triangle = g'(g(< f_\triangle, C_i, C_j >), < C_i, C_j >)$. Both g and g' are trained to minimize the reconstruction loss and the KL-divergence between the prior distribution (in this case, a Gaussian distribution) and the distribution of embeddings $g(< f_\triangle, C_i, C_j >)$. The loss function for C2C-VAE is:

$$\begin{aligned} loss =& ||g'(g(< f_\triangle, C_i, C_j >), < C_i, C_j >) - f_\triangle||^2 \\ &+ KL[g(< f_\triangle, C_i, C_j >), N(0,1)] \end{aligned}$$

Just as a VAE can generate new samples of the original space $L1$. A C2C-VAE can generate new feature differences in $L2'$, which in turn can be used to modify features in $L2$. The modified features can then be recovered as new samples in $L1$. More specifically, A C2C-VAE can be used to generate a new sample s_j of a target class C_j by adapting an existing sample (called as the source sample) using the following procedure:

- Choose a source sample s_i of class C_i in the space $L1$. Get its feature $f(s_i)$ in the space $L2$.
- Sample an embedding $g(< f_\triangle, C_i, C_j >)$ from the Gaussian distribution $N(0,1)$ in space $L3$. Decode this embedding to get a feature difference $f'_\triangle = g'(g(< f_\triangle, C_i, C_j >), < C_i, C_j >)$ in the space $L2'$.
- Apply the feature difference $f'_\triangle$ in $L2'$ to $f(s_i)$ in $L2$, to get the target sample feature $f(s_j) = f(s_i) - f'_\triangle$ in $L2$.
- Decode the target sample feature $f'(f(s_j))$ to the sample space $L1$

This procedure touches each of Boden's three proposed aspects of creativity. It is *exploring* the space $L3$ of difference patterns, *combining* a feature difference with another feature in $L2$, and *transforming* the sample in $L1$.

In Boden's original definition, the boundary between exploratory and transformational creativity is blurred. For this reason, Wiggins (2006) refines Boden's original framework by identifying two rule sets defining the conceptual space (in our previous terminology, this is $L1$, perhaps even $L2$.): the rule set $\mathscr{R}$ that constrains the space and the rule set $\mathscr{T}$ that traverses the space. Transformational creativity can emerge from either transforming $\mathscr{R}$, leading to a new conceptual space, or $\mathscr{T}$, leading to new traversal in the same conceptual space. Under Wiggins' definition, C2C-VAE is applying $\mathscr{T}$-transformation, where samples in the original conceptual space are modified by difference patterns sampled by C2C-VAE.

From the perspective Boden's three criteria of creativity, we hypothesize that C2C-VAE can generate realistic feature differences leading to valuable (within-category) samples, thanks to the regularity offered by both VAE and C2C-VAE. C2C-VAE provides three means for achieving novelty: 1) sampling in the space $L3$, allowing variation in the feature difference; 2) diversifying the source sample and adapting from different samples or classes; 3) sampling in the space $L2$, allowing variation in the feature of the source sample. Our testbed system focuses on the first means for creativity.

Evaluating Creativity for One-Shot Classification Domains

We consider automated evaluation of creativity of generated samples in a classification domain in which a generator can learn to generate samples from data and a classifier can learn to classify samples. We propose a creativity evaluation approach named GOF/TOM (for "generate on few, test on many"). The approach is applicable to multiple forms of generators (e.g. VAE, GAN). Although we describe it in a classification task domain (as we focus on evaluating C2C-VAE), this approach could be applied to regression or other task domains as well.

Given a data set, a class is chosen as the target class. Data samples of that class are called target samples. Some or all target samples are removed from the training set, creating a zero/one/few-shot setting (for simplicity, we will ignore the differences between the three settings and refer them as the one-shot setting), where the target data is not available to the generator in its entirety. Then the generator is trained on the trimmed training set. Meanwhile, a separate model (e.g. classifier) is trained on the untrimmed data set. Because the model sees target samples unknown to the generator, we call it the *oracle*. There can exist multiple oracles serving different purposes, as in our experimental evaluation.

After training of both the generator and the oracle, the generator is used to generate new target samples. The oracle can facilitate the evaluation of the generated sample on the three aspects of creativity (Boden 1991). We assume that the oracle is implemented using deep learning or other techniques enabling extraction of features for the assessment of novelty and/or surprise.

Value The oracle classifies the generated samples. We consider the generator able to generate valuable samples if it can consistently generate samples of the target class. The accuracy of the generator is the percentage of the generated samples falling into the intended class. High accuracy means highly valuable generator.

Novelty The oracle can extract features of the samples into a latent space. The latent space needs to be smooth so that similar samples are close together and different samples are distant from each other. The generated samples are novel if their extracted features show variety. For our experiment, we use the activation of the embedding layer of an VAE as the feature values extracted for each sample. The variance of the features is used as the measure of the variety of the samples generated.

Surprise A generator achieves *surprising* results if it can consistently generate samples of the target class that are unexpected. Existing measures of surprise are surveyed in Franceschelli and Musolesi (2021). Surprise is beyond the scope of this paper and the capability for C2C-VAE to generate surprising samples is left for future research.

Uniqueness and Benefits

Our evaluation method differs from those mentioned earlier in a few ways: The untrimmed data set is not the same as the *inspiring set* (Ritchie 2007) because the data contained are not necessarily creative; The generator learns from the trimmed data set and its knowledge of the target class is deliberately limited. Even if the untrimmed data set is inspiring, its trimmed version may not be; Instead of using a reference set as in Gervás (2011), the oracle classifier is used to evaluate the samples; Unlike the many evaluations such as in Norton, Heath, and Ventura (2010) and Morris et al. (2012), the oracle classifier does not require user rating data or other human assessment of artistry or creativity. It requires only classification labels, which are more widely available.

Methods (Gervás 2011; Morris et al. 2012) that use some (portion of) inspiring set for the evaluation integrate evaluation within the creative system. The creative system filters generated samples by evaluation. However, GOF/TOM estimates is envisioned as a tool for after-the-fact evaluation of the system's performance.

Caveats

In GOF/TOM, the generator is trained in a one-shot setting and then its generated samples are evaluated by an oracle trained using the untrimmed data set. There are three implications: 1) If the task domain is truly one-shot, then the construction of the oracle is impossible, due to the lack of additional data. 2) GOF/TOM may be less suitable for generators with weaker one-shot learning capability. 3) The method assesses value based on classifications by the oracle and novelty based on features generated by the oracle; such features could potentially be used for assessing surprise as well. The usefulness of all three measures depends on how well the oracle has learned from the training data.

A concern for any automated evaluation of creativity is whether it truly captures the important characteristics. We believe that the use of accuracy as a proxy for value, and variance as a proxy for novelty, is reasonable in the domains used for the evaluation. However, these measures may miss important aspects of creativity for some domains. More work is needed on the measures to apply.

Evaluation

We carried out experiments on two data sets, using the GOF/TOM approach to evaluate the creativity of C2C-VAE. The first data set is the MNIST dataset of hand-written digits from 0 to 9. The second is the fashion-MNIST dataset of 10 classes of clothing and accessories. Each of MNIST and fashion-MNIST is provided with predetermined splits for training (60,000 samples) and testing (10,000 samples) data sets; these were used for training and validation. Each sample is a 28x28 grayscale image, associated with a label from 10 classes. The oracle classifier and the VAE are both trained with the full data set. The two data sets are chosen because a traditional VAE can extract features from them.

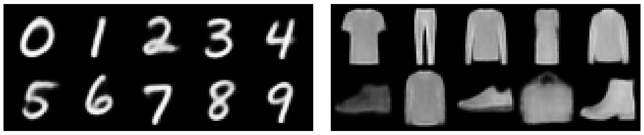
In all experiments, each class is successively chosen as the target class. For comparison with C2C-VAE, we also trained a conditional VAE (CVAE). The CVAE is trained to generate a sample conditioned on an additional parameter controlling the class of the sample generated. During testing, both C2C-VAE and CVAE generate samples of the target class.

System Design

The oracle for value is a resnet18 (not pretrained) network, of which the first layer is replaced with a convolutional layer with $(in\_channels = 1, out\_channels = 64, kernel\_size = (7,7), stride = (2,2), padding = (3,3), bias = False)$, and the last layer is replaced with a linear layer with 10 outputs for classifications.

The VAE follows a standard design. The encoder of the VAE is composite of two consecutive convolutional layers $(out\_channels = c, kernel\_size = 4, stride = 2, padding = 1)$ and $(out\_channels = c * 2, kernel\_size = 4, stride = 2, padding = 1)$, where $c = 64$. A linear layer for mean and another linear layer for log variance follow the convolutional layers and extract a distribution of features from the output of the convolutional layers. A feature is a vector of dimension 32. The decoder of the VAE reverses the design of the encoder: It consists of a linear layer and two consecutive convolutional layers. The input and output dimensions are the reverse of their corresponding encoder layers, other parameters being equal. The VAE is trained with standard reconstruction loss and KL-divergence loss: $Loss_{vae} = loss_{recon} + KL\text{-}divergence$

The VAE's encoder f serves as a feature extractor for the C2C-VAE and also as the oracle for novelty. The resnet18



(a) MNIST (b) Fashion-MNIST

Figure 2: Average Samples Constructed by the VAE

classifier can also extract features but the feature space is not as smooth as that of the VAE.

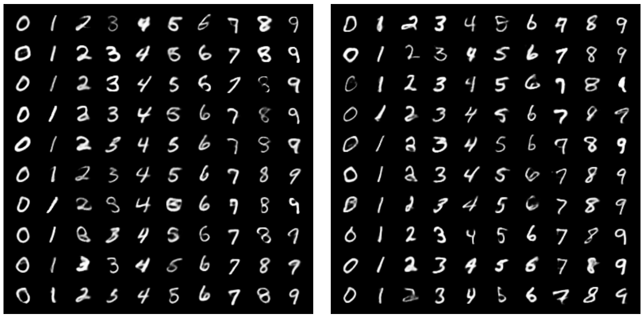
The C2C-VAE has its own encoder and decoder. Given a pair of samples, their features are extracted by the VAE and their class labels are one-hot encoded. The encoder takes the feature difference and the two class labels as input. The input is passed to a fully connected RELU layer with $(out\_features = 32)$. A linear layer for mean and another linear layer for log variance follow and extract a distribution of embeddings, which are of 10 dimensions. The decoder takes an embedding and the two class labels as input. The input is passed to two consecutive linear layers to recover a feature difference similar to the original.

The CVAE has a very similar architecture to the VAE, except it is modified to be conditioned on the class label. Specifically, the encoder and the decoder use the same convolutional layers, but their linear layers that interact with features also take in the class labels as extra inputs.

The C2C-VAE can only generate a new sample by adapting a source sample. We choose an average sample s_{avg} (see Figure 2) from each class C (other than the target class) as the source sample by the following procedure:

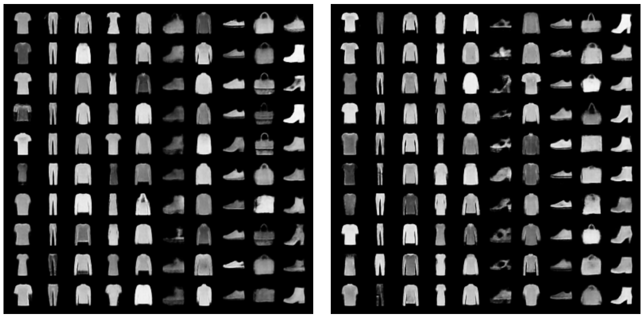
- Select all n samples s_1 - s_n of the class C;
- Calculate the average of their features $avg(\Sigma f(s_i)) = (\Sigma_{i=0}^{n} f(s_i))/n$;
- Use the decoder f' to recover the average sample $f'(avg(\Sigma f(s_i)))$.

In addition to the reconstruction loss, both the C2C-VAE and the CVAE are learning to minimize a KL-divergence loss with a Gaussian distribution ($\mu = 0, \sigma = 1$). This means that they are both trained to project their corresponding input to the Gaussian distribution. Although their inputs and embeddings carry different meanings (The CVAE takes input from $L1$ while the C2C-VAE takes input from $L2'$), we note that their embedding distributions are intended to be the same Gaussian. This also means that we could compare their performance with regard to the standard deviation std of the Gaussian. The experimenter can make either model produce more or less various samples by tuning std, controlling the distribution from which the model is sampling from. The higher std is, the wider the distribution becomes, and the generated samples lose value (accuracy) but gain variety. Note that the C2C-VAE can also introduce additional novelty by altering the source sample, but this comes at a further cost of stability of the results (discussed in the Discussion and Future Work section).



(a) C2C-VAE (b) CVAE

Figure 3: MNIST samples generated by the models trained under normal setting. $std = 1$. Both models demonstrate valuable and various samples.



(a) C2C-VAE (b) CVAE

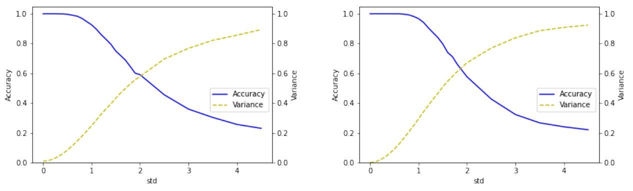
Figure 4: Fashion-MNIST samples generated by the models trained under normal setting. $std = 1$. Both models demonstrate valuable and various samples.

Comparison between the C2C-VAE and the CVAE

Under the Normal Setting Before examining the models under the GOF/TOM setting, we present some results under the normal setting to provide a backdrop for comparison. Under the normal setting, all models are trained on the untrimmed data set. In all figures of generated samples, column j represents the samples generated for class C_j. In all the figures of generated samples by the C2C-VAE, unless otherwise specified, the (i, j), where $i \neq j$, sample is a sample generated by choosing the average sample of class C_i, sampling a feature difference from class i to class j, and applying this feature difference to the chosen sample; Additionally, the (i, i) sample (on the diagonal) is an average sample of class C_i. In all the figures of samples by the CVAE, column j represents samples generated by random sampling in class C_j.

Figures 3 and 4 illustrate that both C2C-VAE and C-VAE produce valuable and varied samples. Because the models have seen various samples of the target class during training, the variety here is not equivalent to novelty (but it will be in GOF/TOM evaluation).

Figure 5 illustrates the tradeoff between value (measured



(a) C2C-VAE

(b) CVAE

Figure 5: Under the normal setting in MNIST, both C2C-VAE and VAE trade off accuracy and variance as std is adjusted. Results are similar for the normal setting in fashion-MNIST

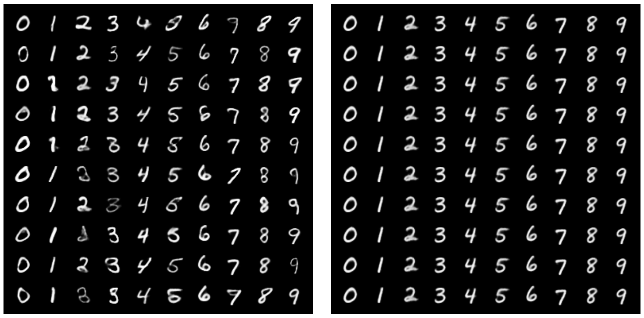
by the accuracy of samples generated judged by the oracle for value) and variety (measured by the variance of the features extracted by the oracle for novelty). Under the normal setting, the two models share similar tradeoffs.

Under the GOF/TOM Setting Our specific implementation of the GOF/TOM setting trims all samples of a target class except one average sample. We choose the average sample to better represent target class. Both CVAE and C2C-VAE are trained with the trimmed data set. During each training epoch, 10% of the training batch is this one-shot sample while 90% is randomly chosen from other samples (CVAE trains on the batch directly while C2C-VAE trains on pairs from the batch). This design counters the imbalanced classes caused by trimming. Therefore the CVAE learns about the target class from only the average sample, while the C2C-VAE learns from pairs of this sample and samples of other classes.

In contrast to the figures of generated samples under the normal setting, the figures presented in this section follow an additional rule: Column j is generated by models which are trained under the one-shot setting, where all samples except the average sample of class j are removed during training. Because the models have only seen a single sample of the target class during training, any variety of generated samples of the target class is equivalent to novelty.

When $std = 1$, the C2C-VAE generates novel—thus creative—samples while the CVAE can only generate similar samples, as shown in Figures 6 and 7.

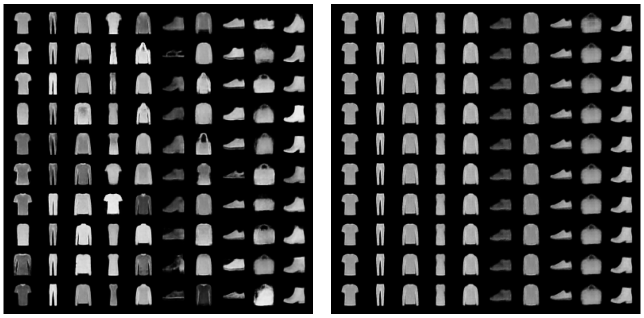
Figures 8 and 9 illustrate the tradeoff between value (measured by the accuracy of samples generated judged by the oracle for value) and novelty (measured by the variance of the features extracted by the oracle for novelty). C2C-VAE exhibits an accuracy and variance tradeoff as std is tuned. For a given level of $variance$, CVAE needs a bigger std change at a bigger cost of $accuracy$ than C2C-VAE. For example, CVAE needs $std = 4$ to gain the same variance as C2C-VAE for $std = 1$ in MNIST, and $std = 4.5$ to gain the same variance as C2C-VAE ($std = 1$) in fashion-MNIST. When std is so high, the quality of the images is very poor, as shown in Figure 10.



(a) C2C-VAE

(b) CVAE

Figure 6: MNIST samples generated by the models trained under one-shot setting. $std = 1$. Intuitively, C2C-VAE generates more creative samples.



(a) C2C-VAE

(b) CVAE

Figure 7: Fashion-MNIST samples generated by the models trained under one-shot setting. $std = 1$. Intuitively, C2C-VAE generates more creative samples.

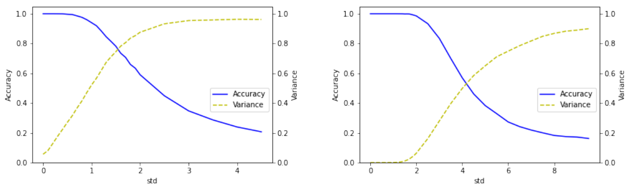
Discussion and Future Work

Conditions for Success of the C2C-VAE Method

The applicability of the C2C-VAE method depends on three conditions: (1) There exists a VAE that can extract features of samples, (2) there exists a C2C-VAE that can extract embeddings of feature differences of two classes, and (3) the generated feature differences can be applied to a chosen sample. Condition (1) depends on properties of VAEs and is beyond the scope of this paper.

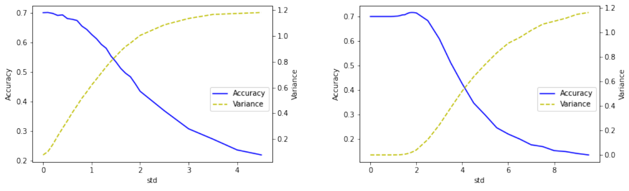
Condition (2) can fail if the feature differences between two classes do not conform to a single pattern. When two classes each have wide distributions, the difference between the two distribution can have very high variation, decreasing effectiveness of the C2C approach (Ye 2018b). For example, drawings in the Quick, Draw! dataset vary considerably within classes. For example, a cat or dog may be drawn with a head only, or with a body, or with limbs and a tail.

Even if condition (2) holds, C2C-VAE can be sensitive to the choice of source sample, causing the failure of condition (3). C2C-VAE can generate creative samples by generating from different source samples, while at the risk of generating bad samples (see Figure 11).



(a) C2C-VAE (b) CVAE

Figure 8: Under the one-shot setting in MNIST, C2C-VAE can trade off accuracy and variance when std is tuned. For a given level of $variance$, CVAE needs a bigger std change at a bigger cost of $accuracy$ than C2C-VAE.



(a) C2C-VAE (b) CVAE

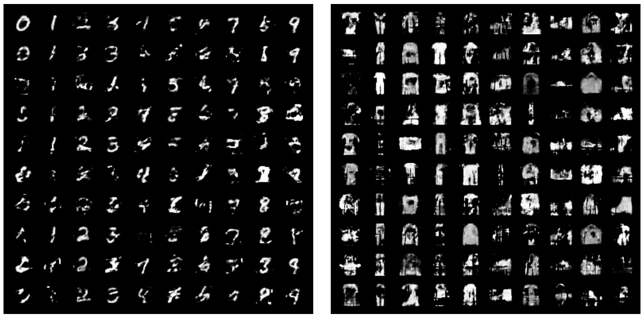
Figure 9: Under the one-shot setting in fashion-MNIST, C2C-VAE can trade off accuracy and variance when std is tuned. For a given level of $variance$, CVAE needs a bigger std change at a bigger cost of $accuracy$ than C2C-VAE.

In the procedure for generating new samples using C2C-VAE, the choice of a source sample s_i and the choice of a feature difference embedding and its subsequently induced feature difference f'_{Δ} are currently two independent choices. There could (and perhaps even should) exist some dependency between the two choices. As a future direction, both conditions (2) and (3) may be resolved by conditioning the C2C-VAE on the source sample.

Relationship to CycleGAN

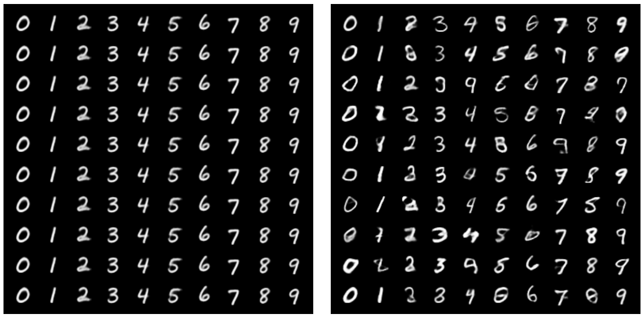
C2C-VAE and CycleGAN are completely different techniques but share many foundational assumptions. CycleGAN assumes a pattern between two classes and trains translation functions (generators) on all possible pairs between two classes to learn it. The reconstruction loss of C2C-VAE (that the feature difference can be recreated) corresponds to the cycle-consistency loss of CycleGAN (that the sample can be recovered). The reconstruction loss of the VAE which C2C-VAE depends upon for feature extraction (a realistic sample can be reconstructed from a feature) corresponds to the adversarial loss of CycleGAN (the recovered sample is realistic). The two models are similar in their foundations, but C2C-VAE works with the space $L2'$ while CycleGAN works with the space $L1$ (and arguably $L2$).

GOF/TOM can benefit from GAN. In its current design, the oracle for value often classifies a generated sample confidently with high activation score even if it is of poor quality by human perception. As a future direction, the oracle might be integrated with a discriminator of GAN to better distinguish poor samples.



(a) MNIST ($std = 4$) (b) Fashion-MNIST ($std = 4.5$)

Figure 10: To gain variance, CVAE requires greatly increases std, sacrificing quality of generated samples



(a) Generated by Modifying Average Samples (b) Generated by Modifying Random Samples

Figure 11: If the source samples are randomly selected, C2C-VAE might generate bad samples. Here $std = 0$, so variance is solely due to the choice of source samples.

Conclusion
Creativity from Inter-Class Patterns

Network-based models provide exciting mechanisms for modeling creativity in AI systems. Existing work on generative methods for creativity can be seen as oriented primarily towards the conceptual space of samples, while C2C-VAE exploits the relationship between samples. If existing approaches look at the foreground of conceptual space $L2$, C2C-VAE looks at the background $L2'$, in order to bring that background to bear as additional information to facilitate creative sample generation in one-shot settings. The presented experiments support that for a creative image generation task, C2C-VAE can achieve high novelty—variance in generated samples—while maintaining accuracy.

Acknowledgements

We acknowledge support from the Department of the Navy, Office of Naval Research (Award N00014-19-1-2655).

Author Contributions

The main tasks and contributing authors are listed below in order of their contribution in each task.

- Conceptualization: Xiaomeng Ye, David Leake, Ziwei Zhao, David Crandall
- Writing: Xiaomeng Ye, David Leake, Ziwei Zhao, David Crandall
- Algorithm Design: Xiaomeng Ye, Ziwei Zhao
- Programming: Ziwei Zhao, Xiaomeng Ye (earlier versions)
- Experimentation and Results: Ziwei Zhao, Xiaomeng Ye
- Review: David Leake, Xiaomeng Ye, David Crandall, Ziwei Zhao

References

Boden, M. A. 1991. *The Creative Mind: Myths and Mechanisms*. USA: Basic Books, Inc.

Broad, T.; Berns, S.; Colton, S.; and Grierson, M. 2021. Active divergence with generative deep learning - A survey and taxonomy. *CoRR* abs/2107.05599.

Draper, S. 2010. Creativity. `https://www.psy.gla.ac.uk/~steve/best/creative.html`. Accessed: 2022-05-08.

Elgammal, A. M.; Liu, B.; Elhoseiny, M.; and Mazzone, M. 2017. CAN: creative adversarial networks, generating "art" by learning about styles and deviating from style norms. *CoRR* abs/1706.07068.

Elhoseiny, M., and Elfeki, M. 2019. Creativity inspired zero-shot learning. *CoRR* abs/1904.01109.

Franceschelli, G., and Musolesi, M. 2021. Creativity and machine learning: A survey. *CoRR* abs/2104.02726.

Gervás, P. 2011. Dynamic inspiring sets for sustained novelty in poetry generation. In Ventura, D.; Gervás, P.; Harrell, D. F.; Maher, M. L.; Pease, A.; and Wiggins, G. A., eds., *Proceedings of the Second International Conference on Computational Creativity, ICCC 2011, Mexico City, Mexico, April 27-29, 2011*, 111–116. computationalcreativity.net.

Ha, D., and Eck, D. 2017. A neural representation of sketch drawings. *CoRR* abs/1704.03477.

Karras, T.; Laine, S.; and Aila, T. 2018. A style-based generator architecture for generative adversarial networks. *CoRR* abs/1812.04948.

Kingma, D. P., and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

LeCun, Y., and Cortes, C. 2010. MNIST handwritten digit database. `http://yann.lecun.com/exdb/mnist/`.

Morris, R. G.; Burton, S. H.; Bodily, P.; and Ventura, D. 2012. Soup over bean of pure joy: Culinary ruminations of an artificial chef. In *ICCC*.

Nobari, A. H.; Rashad, M. F.; and Ahmed, F. 2021. Creativegan: Editing generative adversarial networks for creative design synthesis. *CoRR* abs/2103.06242.

Norton, D.; Heath, D.; and Ventura, D. 2010. Establishing appreciation in a creative system. In *ICCC 2010*.

NVIDIA. 2021. AI artist Helena Sarin. `https://www.nvidia.com/en-us/research/ai-art-gallery/artists/helena-sarin/`. Accessed: 2022-02-17.

Pan, Z.; Yu, W.; Yi, X.; Khan, A.; Yuan, F.; and Zheng, Y. 2019. Recent progress on generative adversarial networks (gans): A survey. *IEEE Access* 7:36322–36333.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17(1):67–99.

Roberts, A.; Engel, J.; Raffel, C.; Simon, I.; and Hawthorne, C. 2018a. Musicvae: Creating a palette for musical scores with machine learning. `https://magenta.tensorflow.org/music-vae`. Accessed: 2022-02-17.

Roberts, A.; Engel, J. H.; Raffel, C.; Hawthorne, C.; and Eck, D. 2018b. A hierarchical latent vector model for learning long-term structure in music. *CoRR* abs/1803.05428.

Sbai, O.; Elhoseiny, M.; Bordes, A.; LeCun, Y.; and Couprie, C. 2018. DeSIGN: Design inspiration from generative networks. *CoRR* abs/1804.00921.

Sohn, K.; Lee, H.; and Yan, X. 2015. Learning structured output representation using deep conditional generative models. In Cortes, C.; Lawrence, N.; Lee, D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Toivonen, H., and Gross, O. 2015. Data mining and machine learning in computational creativity. *WIREs Data Mining and Knowledge Discovery* 5(6):265–275.

Varshney, L. R.; Pinel, F.; Varshney, K. R.; Bhattacharjya, D.; Schörgendorfer, A.; and Chee, Y. 2013. A big data approach to computational creativity. *CoRR* abs/1311.1213.

Wiggins, G. A. 2006. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19(7):449–458. Creative Systems.

Wiggins, G. A. 2021. Creativity and consciousness: Framing, fiction and fraud. In de Silva Garza, A. G.; Veale, T.; Aguilar, W.; and y Pérez, R. P., eds., *Proceedings of the Twelfth International Conference on Computational Creativity, México City, México (Virtual), September 14-18, 2021*, 182–191. Association for Computational Creativity (ACC).

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR* abs/1708.07747.

Ye, X.; Leake, D.; Huibregtse, W.; and Dalkilic, M. 2020. Applying class-to-class siamese networks to explain classifications with supportive and contrastive cases. In *International Conference on Case-Based Reasoning*, 245–260. Springer.

Ye, X.; Leake, D.; Jalali, V.; and Crandall, D. J. 2021. Learning adaptations for case-based classification: A neural network approach. In Sánchez-Ruiz, A. A., and Floyd, M. W., eds., *Case-Based Reasoning Research and Development*, 279–293. Cham: Springer International Publishing.

Ye, X. 2018a. The enemy of my enemy is my friend: Class-to-class weighting in k-nearest neighbors algorithm. In *Proceedings of the Thirty-First International Florida Artificial Intelligence Research Society Conference, FLAIRS 2018*, 389–394.

Ye, X. 2018b. The enemy of my enemy is my friend: Class-to-class weighting in k-nearest neighbors algorithm. In *FLAIRS Conference*.

Zhu, J.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR* abs/1703.10593.

Simultaneous Multiple-Prompt Guided Generation Using Differentiable Optimal Transport

Yingtao Tian
Google Brain
Tokyo, Japan
alantian@google.com

Marco Cuturi
Google Brain (currently at Apple)
Paris, France
cuturi@apple.com

David Ha
Google Brain
Tokyo, Japan
hadavid@google.com

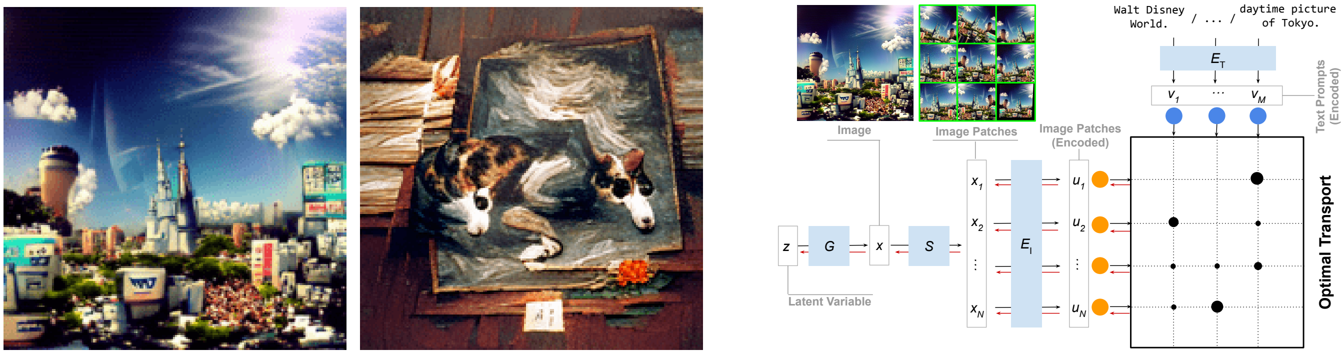
Abstract

Recent advances in deep learning, such as powerful generative models and joint text-image embeddings, have provided the computational creativity community with new tools, opening new perspectives for artistic pursuits. *Text-to-image synthesis* approaches that operate by generating images from text cues provide a case in point. These images are generated with a latent vector that is progressively refined to agree with text cues. To do so, patches are sampled within the generated image, and compared with the text prompts in the common text-image embedding space; The latent vector is then updated, using gradient descent, to reduce the mean (average) distance between these patches and text cues. While this approach provides artists with ample freedom to customize the overall appearance of images, through their choice in generative models, the reliance on a simple criterion (mean of distances) often causes mode collapse: The entire image is drawn to the *average* of all text cues, thereby losing their diversity. To address this issue, we propose using matching techniques found in the optimal transport (OT) literature, resulting in images that are able to reflect faithfully a wide diversity of prompts. We provide numerous illustrations showing that OT avoids some of the pitfalls arising from estimating vectors with mean distances, and demonstrate the capacity of our proposed method to perform better in experiments, qualitatively and quantitatively.

Introduction

The computational creativity community has been at the forefront of engaging with recent advances in deep learning, adopting early on generative models that are able to produce high-quality text and images. Such models offer varying degrees of realism and control to the artist, enabling the generation of results with artistic value. Recent advances have brought forward models that can produce images from natural language prompts, using pre-trained image generative models guided by text descriptions (Radford et al. 2021). The computational creativity community has seized this opportunity, has shared large bodies of code (Burton-King 2021; Murdock 2021a) and generated a large body of artwork, some of which has been curated online (Snell 2021; Murdock).

These tools are favoured by artists because they can shape generation in various ways: For instance, a relevant genera-



(a) Generated images from *two-prompts* using our method. (Left) "*Walt Disney World.*" and "*a daytime picture of Tokyo.*" (Right) "*A painting of cat.*" and "*A painting of dog.*".

(b) The architecture of our work. Iteratively, the loss is computed forward (marked by $\rightarrow$) and the gradient is calculated backward (marked by $\leftarrow$) to update the latent variable z.

Figure 1: Our method illustrated with generated images and the architecture. In contrast, the existing method would fail with these two-prompts, producing images with less diverse features (left) or a painting with much different art style than single prompt (right). This is because the existing method of taking the mean cannot treat different parts of the image separately, and vector arithmetic in the latent space introduces uncontrollable changes in the semantics. Detailed analysis can be found in text. All figures in this paper are *generated* using pre-trained CLIP and VQGAN models, both publicly released under MIT license.

tive model can be used in that the style of pieces of art that can be produced can be efficiently guided by selecting a relevant generative model G. While this degree of freedom is useful, little has changed on how text prompts are handled in that pipeline:

Images can be generated using the following pipeline: The user supplies a generative model G and a text prompt t. An initial latent vector z is sampled randomly; the fit between its corresponding image $x = G(z)$ and the desired prompt t is quantified using their distance in a common CLIP embedding space; In order to minimize that distance, z is updated iteratively using gradient steps. Because both the CLIP embedding and G are differentiable, gradients for these distances are obtained using automatic differentiation.

In practice, a few more tricks are needed to produce convincing images. To accommodate the important artistic requirement that multiple concepts appear in images, several text prompts are allowed, but are pre-aggregated in embedding space to result in a composite prompt vector. Next, rather than consider the entire image against that composite prompt vector, several patches with random size, orientation and placement are sampled within the image x, and are then compared with the composite prompt, before these distances are averaged to form the overall loss.

These tricks rely therefore on aggregations: the mean of various prompt embeddings is used to define a single target prompt, and the various distances of all patches to that target are also reduced to their average. We argue, and we show later in the paper, that this reliance on averages can cause several issues, causing notably generated images to have parts that are uniformly closer to all prompts, thus defeating the original motivation of using multiple prompts to obtain artistic images with diverse objects. Another important drawback of averaging prompt embeddings is that it can potentially introduce uncontrollable changes in semantics, with a mean prompt embedding falling in a region of the embedding space with no corresponding meaning.

We propose to address this issue by treating the embedded patches of generated image and texts as vectors sampled from two probability distributions, and to use computational optimal transport (OT) (Peyré and Cuturi 2019) to find the best matching between them. As its name suggests, OT tries to find the minimal total effort required to “move” all patches towards texts, using the pairwise distance as the cost for measuring said effort. OT brings two advantages over simply taking the mean: (1) Since patches are randomly sampled, it encourages the intrinsic diversity *inside* a single generated image. (2) OT does not involve vector arithmetic in the latent space, sidestepping issues that may arise from the non-existing semantic of a mean prompt vector. Concretely, we use Sinkhorn’s Algorithm (Cuturi 2013; Séjourné et al. 2019) for the matching, in a way that is efficient and, most importantly, differentiable using OTT-JAX (Cuturi et al. 2022). Such differentiability is crucial to allow the computation of gradient all the way back to z.

Bringing all pieces together, our proposed use of OT enables the generation of images that are diverse and without the issue of unwanted extra semantics, as demonstrated empirically in the paper. Furthermore, since our proposed method only changes how pairs of (patch, prompt) distances are recombined, it is orthogonal to other existing parts of the pipeline, and consists, implementation-wise, in a simple drop-in replacement of mean operations by optimised matchings (incidentally, taking means can be interpreted as the most naive approach conceivable to match pairs). We start this paper with a background section, needed to detail next our methodology, which is illustrated and validated in various experiments that showcase its performance, and explain why it is able to solve several issues arising from an over-reliance on mean distances and mean prompt embeddings.

Background

In this section, we review two pillars of our work, *prompt-guided image generation* and *differentiable optimal transport*. We argue in this paper that combining both is crucial to address issues we observe in existing generation methods.

Prompt Guided Image Generation

A notable trend in the field of computational creativity is to guide image generation using natural language as prompts. These *text-to-painting* synthesis tools allow artists to specify the content of a painting using prompts from natural languages. This text-driven generation has revolutionized the computational generation of artworks, as evidenced in online curated collections (Snell 2021; Murdock). These advances are made possible by combining two innovations from deep learning:

Powerful image generative models. Such models include recent generative adversarial networks (GANs) (Karras, Laine, and Aila 2019; Karras et al. 2020; Karras et al. 2021), variational autoencoders (van den Oord, Vinyals, and Kavukcuoglu 2017) and diffusion models (Ho, Jain, and Abbeel 2020; Song, Meng, and Ermon 2020; Nichol and Dhariwal 2021; Dhariwal and Nichol 2021), that can produce images with high fidelity and diversity. Formally, this process can be denoted as $x = G(z)$ where the generative model $G : \mathbb{R}^{d_z} \to \mathbb{R}^{h\times w\times 3}$ converts a latent space variable $z \in \mathbb{R}^{d_z}$ to an RGB image of height h, weight w and 3 color channels. $x \in \mathbb{R}^{h\times w\times 3}$. z could be further manipulated to allow generating more suitable x (Li, Jin, and Zhu 2021), allowing artist to control the generation of artworks that fall in desired genres (Jin et al. 2017).

Joint modeling of images and natural language. This idea has been long in the making (Thomee et al. 2016; Li et al. 2017), but only recently given a convincing implementation thanks to progress in natural languages modeling (Raffel et al. 2019; Brown et al. 2020), and notably the ability to embed jointly images and text so well that the need for task-specific fine-tuning is eliminated, as shown in CLIP (Radford et al. 2021). CLIP provides two jointly-trained, differentiable encoders, $E_{\mathrm{I}} : \mathbb{R}^{h\times w\times 3} \to \mathbb{R}^d$ and $E_{\mathrm{T}} : \mathcal{T} \to \mathbb{R}^d$, for image and text respectively. We do not further elaborate the domain of text $\mathcal{T}$ as it is not the focus of this work. Formally, given an image x and a text t, and a distance function $D : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ the encoded image $u = E_{\mathrm{I}}(x)$ and the encoded text $v = E_{\mathrm{T}}(t)$ are in a common comparable space $\mathcal{U} = \mathbb{R}^d$, and $D(u, v)$ measures the similarity between x and t. In the case of CLIP that is trained with cosine distance, practically D could be chosen as cosine distance or geodesic distance, both effectively measuring the angle between the two vectors and being trivially differentiable. Ideally, text-driven image generation is now feasible by iteratively adjusting the latent space vector z, to minimize $D(u, v)$, the distance between the encoded image $x = G(z)$ and encoded user-specific prompt t. As G, E_{I} and D are differentiable, z could be updated using gradient Descent: $z \leftarrow z - \gamma \nabla_z F(z)$ where $\nabla_z F$ is the gradient of F defined as $F(z) = D(E_{\mathrm{I}}(G(z)), E_{\mathrm{T}}(t))$ and γ is a learning rate.

Using a distance from a single image to a single prompt is usually too restrictive. Therefore, and in practice, the distance is computed over pairs of multiple images and texts as follows: On the image side, n patches (a.k.a. cutouts. We use these two terms interchangeably), which we denote as $x_1, \cdots, x_n = S(x)$ are randomly sampled from image x in the fashion of image data augmentation (Shorten and Khoshgoftaar 2019). We assume $x_i \in \mathbb{R}^{h \times w \times 3}$ still holds since we can trivially add a resizing step at the end of augmentation. This practice serves as a regularizer to ensure numerical stability and avoid fitting into regions of z where G has bad support. On the text side, m text prompts, denoted as $t_1, \cdots, t_m$, are often considered, which allows artists to explore the possibilities of art by combining multiple texts as directions. Again, they are encoded accordingly, giving $u_1, \cdots, u_n : u_i = E_{\mathrm{I}}(x_i) \in \mathbb{R}^d$ and $v_1, \cdots, v_m : v_j = E_{\mathrm{T}}(t_j) \in \mathbb{R}^d$. These pairwise distances are then combined to form a loss, which is

$$F(z) = \mathrm{Mean}_D(z) \overset{\text{def}}{=} \frac{1}{mn} \sum_{1 \le i \le n, 1 \le j \le m} D(u_i, v_j), \quad (1)$$

and thus the gradient $\nabla_z F$ reads

$$\nabla_z F = \left(\sum_{1 \le i \le n} \frac{\partial \mathrm{Mean}_D}{\partial u_i} \frac{\partial u_i}{\partial x_i} \frac{\partial x_i}{\partial x} \right) \frac{\partial x}{\partial z} \quad (2)$$

where

$$\begin{aligned} \frac{\partial \mathrm{Mean}_D}{\partial u_i} &= \frac{1}{nm} \sum_{1 \le j \le m} \frac{\partial D(u_i, v_j)}{\partial u_i} \\ \frac{\partial u_i}{\partial x_i} &= \nabla_x E_{\mathrm{I}}(x_i), \quad \frac{\partial x}{\partial z} = \nabla_z G(z) \end{aligned} \quad (3)$$

and $\partial x_i / \partial x$ is defined as long as the random sampling is differentiable w.r.t. the input image x which is often the case of data augmentations. This framing of text-driven generation has been applied to different generators G, yielding a variety of artistic results: using unconditional GAN generation, like BigGAN (Murdock 2021a), VQGAN (Burton-King 2021) and SIREN (Murdock 2021b); conditional generation using GAN, such as StyleCLIP (Patashnik et al. 2021), that enables editing existing images. In addition to GANs, it can also be applied to Diffusion models (Crowson 2021; Kim and Ye 2021; Nichol et al. 2021).

Differentiable Optimal Transport

Optimal transport (OT), as its name suggests, can be understood as finding an efficient way to 'move' or 'transport', the mass from a probability distribution to another distribution. We borrow notations from the survey book (Peyré and Cuturi 2019) and focus on one of the canonical OT formulations, one that was proposed in (Kantorovich 1942). A discrete measure with weights a on locations $u_1, \cdots, u_n$ would be denoted as $\alpha = \sum_{1 \le i \le n} a_i \delta_{u_i}$, where notation δ_u stands for a Dirac mass at location u. Similarly, for weights b on locations $v_1, \cdots, v_m$ we have $\beta = \sum_{1 \le j \le m} b_j \delta_{v_j}$. A possible way to map a discrete measure α onto β, given a cost matrix $\mathbf{C} \in \mathbb{R}_+^{n \times m}$, can be represented with a coupling matrix $\mathbf{P} \in \mathbb{R}_+^{n \times m}$, where the amount of mass transported from the i-th location in α to j-th location in β is stored as $\mathbf{P}_{i,j}$. The set of admissible couplings, $\mathbf{U}$, is defined through a and b as

$$\mathbf{U}(a, b) \overset{\text{def}}{=} \left\{ \mathbf{P} \in \mathbf{R}_+^{n \times m} : \sum_j \mathbf{P}_{i,j} = a, \quad \sum_i \mathbf{P}_{i,j} = b \right\},$$

These row- and and column-sum constraints for P indicate that the entire mass from α is indeed transported to β. Kantorovich's problem of interest is

$$\mathrm{L}(a, b, \mathbf{C}) \overset{\text{def}}{=} \min_{\mathbf{P} \in \mathbf{U}(a,b)} \langle \mathbf{C}, \mathbf{P} \rangle \overset{\text{def}}{=} \sum_{i,j} \mathbf{C}_{i,j} \mathbf{P}_{i,j},$$

which can be solved using linear programming, notably network flow solvers. The linear programming route, while well established, has a few drawbacks: it is slow, with an unstable solution. A possible workaround is to add an entropic regularization term, where the entropy of P reads $\mathbf{H}(\mathbf{P}) \overset{\text{def}}{=} -\sum_{i,j} (P)_{i,j} (\log(\mathbf{P}_{i,j}) - 1)$. The regularized problem reads:

$$\mathrm{L}^{\epsilon}(a, b, \mathbf{C}) \overset{\text{def}}{=} \min_{\mathbf{P} \in \mathbf{U}(a,b)} \langle \mathbf{C}, \mathbf{P} \rangle - \epsilon \mathbf{H}(\mathbf{P}).$$

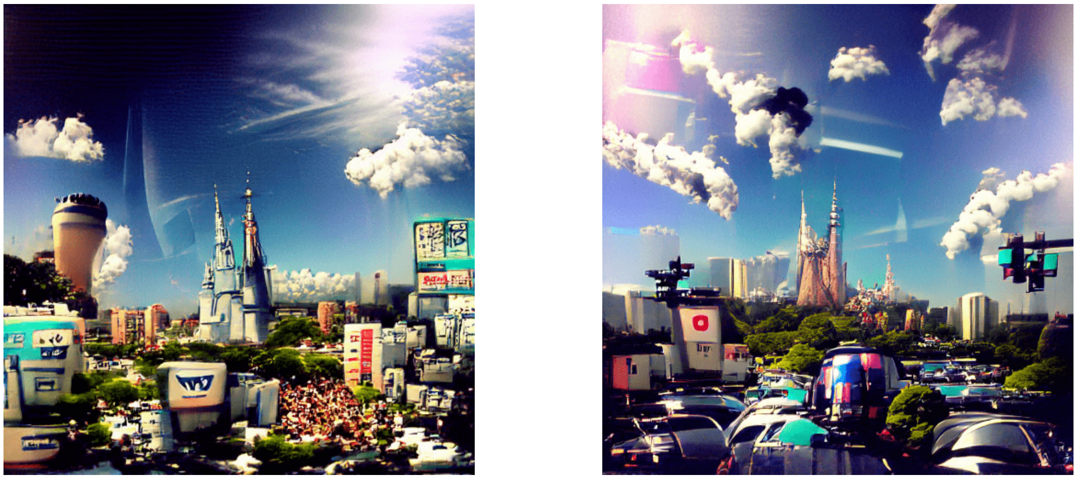
This regularization has several practical virtues: the regularized problem can be solved efficiently with Sinkhorn's Algorithm (Cuturi 2013; Séjourné et al. 2019), a fast iterative algorithm that only uses matrix-vector arithmetic. Another advantage, equally important in our setting, is that this approach, as implemented in OTT-JAX (Cuturi et al. 2022) results in fully differentiable quantities. Namely, assume that the cost matrix $\mathbf{C}$ is provided in the form of a differentiable function resulting in entries $\mathbf{C}_{i,j} \overset{\text{def}}{=} \mathbf{C}(u_i, v_j)$. Then the gradient of L^{ϵ} w.r.t. u_i exists and is defined everywhere:

$$\forall i, 1 \le i \le n, \left\| \frac{\partial \mathrm{L}^{\epsilon}}{\partial u_i} \right\| < \infty. \quad (4)$$

Not that the optimal solution P^{ϵ} corresponding to L^{ϵ} can also be differentiated w.r.t. any of the relevant inputs, using the implicit function Theorem (Krantz and Parks 2002), as proposed in OTT-JAX (Cuturi et al. 2022), but this is not used in this paper because we rely on Danskin's Theorem (Danskin 1966) (a.k.a Envelope Theorem) to differentiate L^{ϵ} w.r.t. $\mathbf{C}$.

Methodology

Our motivation comes from the concern arising from using an averaged loss Mean_D. By focusing on means, all sampled



(a) Our Proposed Method (OT)

(b) Baseline (Mean)

Figure 2: The generated image from two prompts: "Walt Disney World." and "daytime picture of Tokyo." Compare with the baseline, our methods generates images with better diversity (Disney-like architecures vs. city scense) while blending them well.

patches are encouraged to move uniformly to the mean of all prompts. This undermines the very motivation of introducing multiple prompts, which is to allow artists to obtain spatial diversity in the generated images, with various areas reflecting the diversity prompts. Furthermore, taking the mean in the embedding space introduces gradients in unwanted directions. Since the locations in the embedding space are associated with semantics, doing so may introduce uncontrollable, redundant semantics. To make things worse, the mean arithmetic effectively assumes an Euclidean space, which is inconsistent to the CLIP model that is trained with cosine distance in the embedding space.

To address these issues, it is possible to devise an arithmetic in non-Euclidean Space. However, finding a proper choice that works well with the rest of pipeline is not trivial and warrants a separate study. Instead we propose to eliminate the undesired simplifications brought by mean arithmetics, to replace Mean_D in Equation 1 with an optimal transport loss,

$$F = \mathrm{L}^{\epsilon}(a, b, [D(u_i, v_j)]_{i,j}) \tag{5}$$

where $a_i = 1/n$ and $b_j = 1/m$, and the cost matrix $\mathbf{C}$ is populated with pairwise distance D evaluations. Now, the gradient $\nabla_z F$ reads

$$\nabla_z F = \left(\sum_{1 \leq i \leq n} \frac{\partial \mathrm{L}^{\epsilon}}{\partial u_i} \frac{\partial u_i}{\partial x_i} \frac{\partial x_i}{\partial x} \right) \frac{\partial x}{\partial z}. \tag{6}$$

Comparing with Equation 2, the only different term is $\frac{\partial \mathrm{L}^{\epsilon}}{\partial u_i}$ which is also defined as in Equation 4. Along with other terms (see Equation 3), all terms are defined, and thus we know that $\nabla_z F$ is also well-defined and can be used in the iteratively updating of z:

$$z \leftarrow z - \gamma \nabla_z F$$

In doing so, the above mentioned issues are solved for the following reasons:

OT Treats different patches differently. As OT matches patches and text prompts, it naturally introduces a distinct treatment of patches according to their distances to text prompts. As the patches are randomly sampled, it encourages the intrinsic diversity *inside* a single generated image.

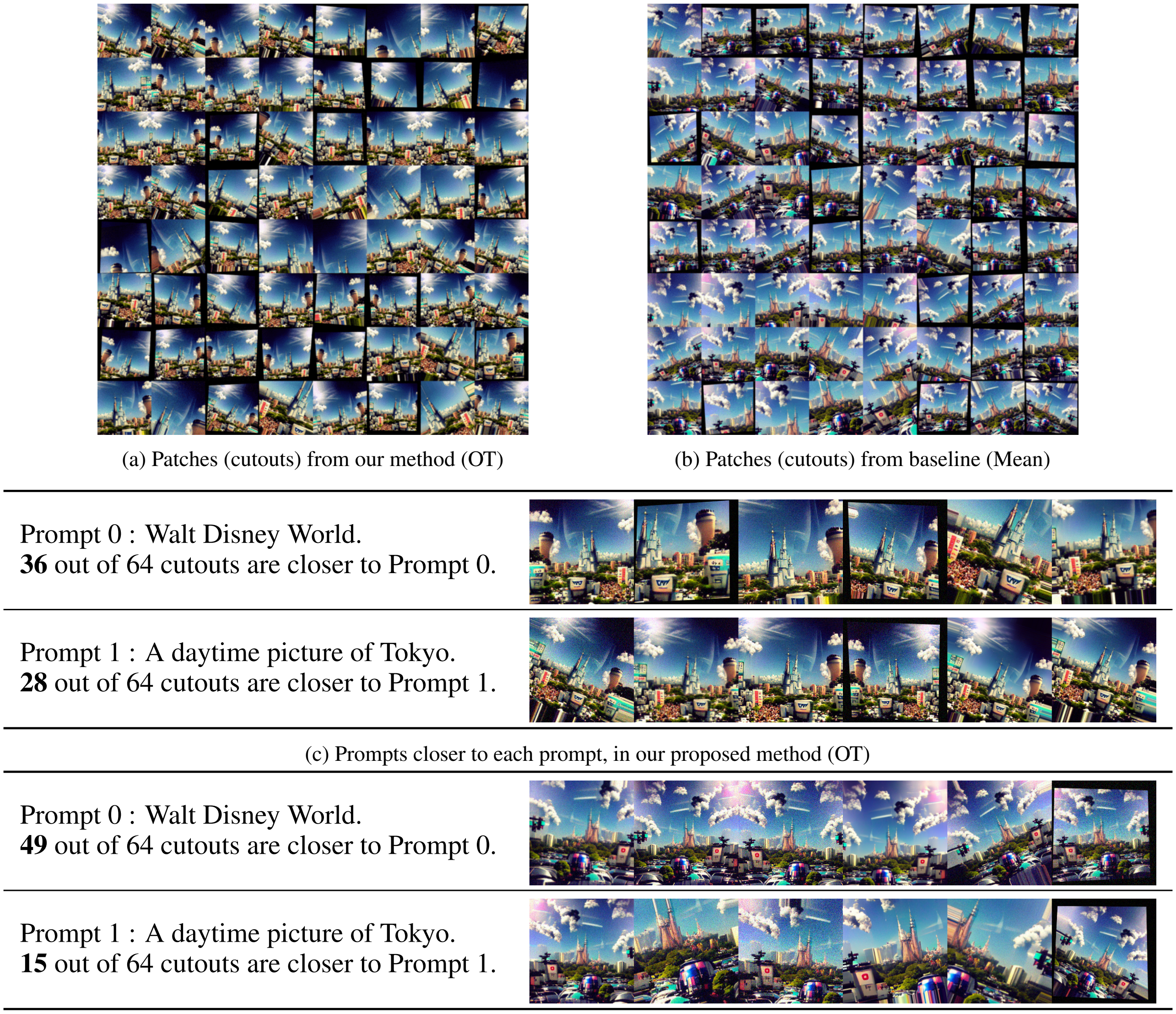
OT does not involve arithmetic in the latent space. OT relies on distances, but does not use averages in embedding spaces. Therefore it does not produce synthetic prompts in embeddings space that may not correspond to semantics. Furthermore, OT is agnostic to how distances are defined: any distance, other than cosine distance or geodesic distance, could be used to populate matrix $\mathbf{C}$.

Experiments

In this section, we highlight a few possibilities brought forward by using our methodology when handling multiple text prompts. Due to the creative nature of text-to-image synthesis, there is no standard measuring stick, such as classification accuracy, to provide a simple comparison between methods. Nevertheless, we consider a few tasks that can help us gain insight into the novelty, the properties and the behavior of our method. We consider:

Generated Image. Naturally the foremost task is to show the generated image x with multiple text prompts $t_1, \cdots, t_m$. In this task, we focus on whether the generated image represents the text prompts in a way that is distinctive and subjectively recognized by human viewers.

Patches (Cutouts) from Generated Images. Our method improves the diversity of patches through increasing the correlation between the distribution of randomly sampled patches and multiple text prompts, as we identify as a source of issues from existing practices. In this task, we show the patches and organize them by text prompt. Formally, we show the n patches $x_1, \cdots, x_n$ sampled from x, and group x_i by $j^* = \arg\min_j D(u_i, v_j)$, the closest text prompt in



(a) Patches (cutouts) from our method (OT)

(b) Patches (cutouts) from baseline (Mean)

Prompt 0 : Walt Disney World.
36 out of 64 cutouts are closer to Prompt 0.

Prompt 1 : A daytime picture of Tokyo.
28 out of 64 cutouts are closer to Prompt 1.

(c) Prompts closer to each prompt, in our proposed method (OT)

Prompt 0 : Walt Disney World.
49 out of 64 cutouts are closer to Prompt 0.

Prompt 1 : A daytime picture of Tokyo.
15 out of 64 cutouts are closer to Prompt 1.

(d) Prompts closer to each prompt, in baseline (Mean)

Figure 3: The cutouts (patches) from generated images in Figure 2, for both our proposed method (OT) and the baseline. We show in (a) and (b) the sampled patches. Then in (c) and (d) we group these patches by the closer (measure by D) prompt they are to. Due to space constraints, we only show the number of each group and six patches that are mostly closet.

the embedding space.

Tangent of Patches (Cutouts) on Cost Plane. We identify the issue materialize in the way gradient information is pass from F back to patches, which is $\partial \mathrm{Mean}_D / \partial u_i$ part in Equation 2, and propose to use $\mathrm{L}_\mathrm{C}^\epsilon$ such that the $\partial \mathrm{L}_\mathrm{C}^\epsilon / \partial u_i$ part in Equation 6, is better.

To quantitatively qualify such property, a few extra deliberations are needed. Concretely, we first define

$$\phi(u_i) : \mathbb{R}^d \to \mathbb{R}^m \overset{\text{def}}{=} [D(u_i, v_1), \cdots, D(u_i, v_m)],$$

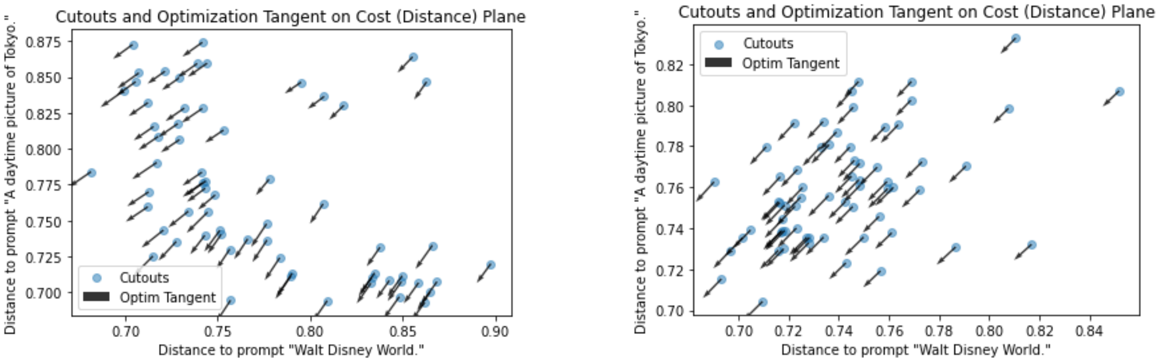
which is by definition a differentiable mapping from the aforementioned embedding space $\mathbb{R}^d$ to $\mathbb{R}^m$, a m-d space of distances to prompts where the j-th element is the distance to prompt j. As $\partial \mathrm{L}_\mathrm{C}^\epsilon / \partial u_i \in T_{u_i}$ (the tangent space of $\mathbb{R}^d$ at u_i), the pushforward by ϕ at u_i is defined as $d\phi : T_{u_i}\mathbb{R}^d \to T_{\phi(u_i)}\mathbb{R}^m$ such that when applied to the gradient,

$$w_i = d\phi(\partial \mathrm{L}_\mathrm{C}^\epsilon / \partial u_i) \tag{7}$$

is in the tangent space of $\mathbb{R}^m$. Intuitively, w_i is a m-dimensional vector whose j-th element denotes the component of gradient that moves the i-th patch towards the j-th text prompt.

Comparing our Method with the Baseline for Two Prompts Setting

In this experiment, we focus on a scenario with $M = 2$ prompts, "*Walt Disney World.*" and "*daytime picture of*



(a) Our method (OT)

(b) Baseline (Mean)

Figure 4: Tangent after pushforward of the gradients on each patch (cutout) in the embedding space to the cost plane. Each blue dot is a patch (cutout), and intuitively, its coordinate shows the distance to one of two prompts, while its arrow shows the force of gradient that pushes it towards the prompts. On the left side, in our method the force of gradient pushes patches to prompts with different "mix ratio", promoting the intrinsic diversity in the generated image from which patches are sampled. On the right side, in the baseline all patches are pushed for the same mix of prompts, thus leading to less diversity. Formally, the exact form and motivation for the tangent could be found mathematically in Equation 7 and its discussions.

Tokyo." We compare two models, our proposed approach with Optimal Transport (Equation 5) and the baseline using Mean (Equation 1), with the purpose of investigating the behavior of these methods and the difference made by our approach. We keep all other configurations the same. Namely, we use a pre-trained VQGAN (Esser, Rombach, and Ommer 2021) on Imagenet dataset, $N = 64$ randomly sampled patch, and 1000 iterations of updating z. We organize the conducted tasks as explained before.

Generated Image and *Patches (Cutouts) from it.* In Figure 2 we show the generated image from both methods. Also in Figure 3 we show the patches (cutouts) sampled from the generated images at the end of all iterations.

We observe that OT helps generate images where patches (cutouts) are more balanced (**36/28** vs **49/15**). Furthermore, OT's results are more diverse for two prompts. For OT, patches close to "Walt Disney World." are more like closeups and patches close to "A daytime picture of Tokyo." are mostly zoomed-out. As patches are randomly done, it reflects the intrinsic property of generated images.

Tangent of Patches (Cutouts) on Cost Plane. We pushforward gradients on the patches' embedding space to this cost plane, as explained in Equation 7, and show the results in Figure 4. We observe that our method using OT clearly shows that the positions in the cost plane reveal negative correlation, which means that different parts of the generated images are successfully encouraged to provide contribution to the similarities to different promoters. This is cross-verified by the "fan out" of tangents pushed forward to the cost plane, which shows the divergent gradients providing patch-specific directions in updating. In contrast, baseline methods are simply learning to be the mean of two prompts' embeddings, as the tangents show the uniformed gradient direction which does not distinguish between different prompts.

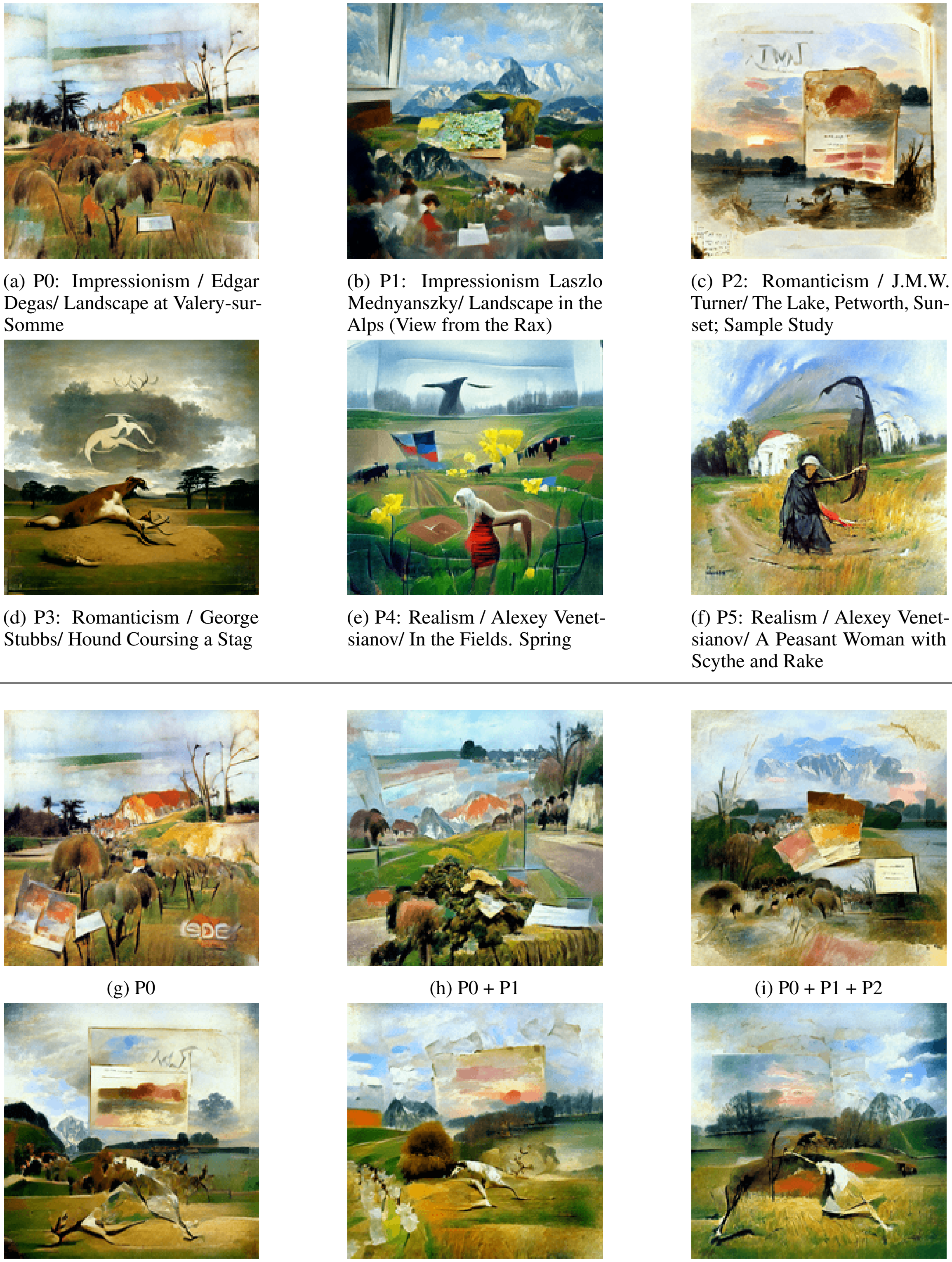
Our Method's Behavior with Multiple Prompts

Having comparing our OT-based method with the baseline on the two prompts setting, we shift our focus to the scenario where our method is applied to multiple prompts. As this is we designed our method to expose fine differentiation among prompts, it becomes interesting to investigate such behavior when the number of prompts increases. In doing so, we consider totally $M = 6$ prompts, numbered from P0 to P5:

- P0: Impressionism / Edgar Degas/ Landscape at Valery-sur-Somme
- P1: Impressionism Laszlo Mednyanszky/ Landscape in the Alps (View from the Rax)
- P2: Romanticism / J.M.W. Turner/ The Lake, Petworth, Sunset; Sample Study
- P3: Romanticism / George Stubbs/ Hound Coursing a Stag
- P4: Realism / Alexey Venetsianov/ In the Fields. Spring
- P5: Realism / Alexey Venetsianov/ A Peasant Woman with Scythe and Rake

and as the prompts suggest, we use a pre-trained VQGAN on WikiArt dataset consisting mostly of paintings. The purpose is to both show that our method could be applied to generative models trained from different genre data, and also that the painting allows easier qualitative comparison of both objects and artistic styles. As the same setting mentioned above, $N = 64$ randomly sampled patch, and 1000 iterations of updating are used. We conduct tasks as explained before.

Generated Image. In Figure 5, we show in the first group the generated images corresponding to these prompts individually, and in the second group the generated images by combining prompts using our proposed method. We observe that our method is capable of composing the instructions from several prompts, in terms of styles and objects, into the same canvas.



(a) P0: Impressionism / Edgar Degas/ Landscape at Valery-sur-Somme

(b) P1: Impressionism Laszlo Mednyanszky/ Landscape in the Alps (View from the Rax)

(c) P2: Romanticism / J.M.W. Turner/ The Lake, Petworth, Sunset; Sample Study

(d) P3: Romanticism / George Stubbs/ Hound Coursing a Stag

(e) P4: Realism / Alexey Venetsianov/ In the Fields. Spring

(f) P5: Realism / Alexey Venetsianov/ A Peasant Woman with Scythe and Rake

(g) P0

(h) P0 + P1

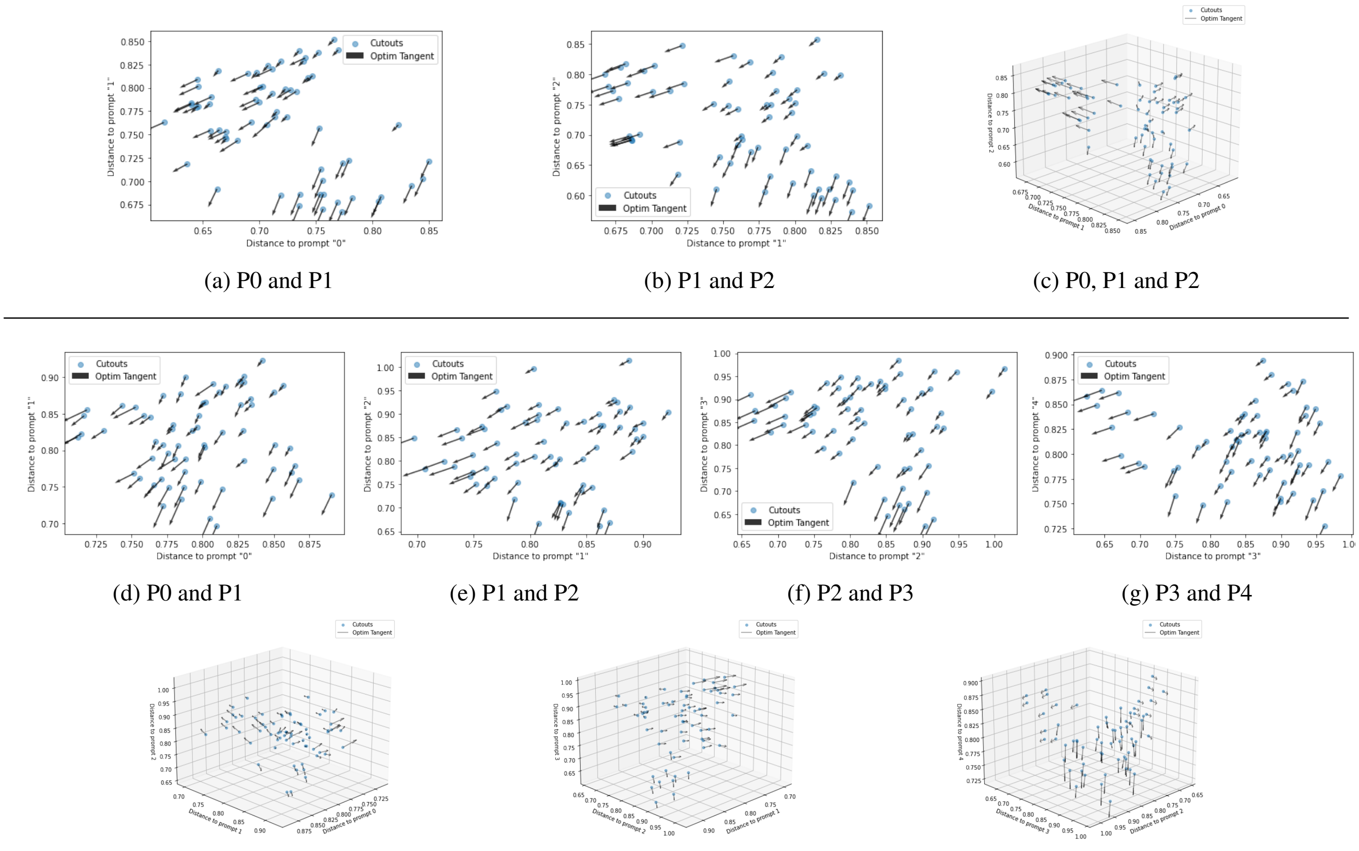
(i) P0 + P1 + P2

(j) P0 + P1 + P2 + P3

(k) P0 + P1 + P2 + P3 + P4

(l) P0 + P1 + P2 + P3 + P4 + P5

Figure 5: The generated images from multiple (6) prompts, labeled P0 to P5. (a) - (f): The first group of 6 images are generated using each one prompt respectively, as a controlling group. (g) - (i): The second group of 6 images are the generated images with multiple (1 to 6) prompts respectively from our proposed method, each one of which using a combination of multiple problems specified in the caption.



(a) P0 and P1

(b) P1 and P2

(c) P0, P1 and P2

(d) P0 and P1

(e) P1 and P2

(f) P2 and P3

(g) P3 and P4

(h) P0, P1 and P2

(i) P1, P2 and P3

(j) P2, P3 and P4

Figure 6: Tangent, representing the gradients on patches (cutouts) after they are pushed forward to Cost Plan. The first group is for the generation with 3 prompts and the second group is for the generation with 6 prompts, showing in 2D and 3D slices.

Tangent of Patches (Cutouts) on Cost Plane. In Figure 6, we show that the good behavior on tangent remains even for multiple prompts. This means that our method is capable of guiding generating images that are diverse in its contents w.r.t. multiple prompts.

Conclusion and Future Work

In this paper we discuss the problem in dealing with multiple text prompts in the setting of text-driven image generation for computational creativity setting. We then propose to address the issue using OT (Optimal Transport) between sampled patches in the generated image and multiple text prompts, and show its theoretical motivation and quantitative and qualitative empirical results highlighting the advantage brought by our proposed method.

One of the advantages in our method is that it is in theory orthogonal to other parts in the whole text driven image generation pipeline, as we show primarily that it works for VQGAN trained on several datasets. We envision that future work would investigate leveraging our proposed method to other drastically different forms of generative method, such as diffusion models. Another possible future direction may principally study the combination of optimal transport and adaptive sampling where in our proposed work only random sampling is used for simplicity.

Author Contributions

Yingtao: Ideated the problem and the method, conducted experiments, drafted the paper.

Marco Cuturi: Provided advises, helped designing the method / experiments, helped drafting the paper.

David Ha: Provided advises and gave feedback / suggestions for the whole work, helped drafting the paper.

Acknowledgement

The authors thank Yujin Tang, Jerry Li, Mathieu Blondel for their valuable comments and suggestions.

References

[Brown et al. 2020] Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

[Burton-King 2021] Burton-King, S. 2021. Introduction to vqgan+clip. https://bit.ly/3rcedh4.

[Crowson 2021] Crowson, K. 2021. Clip guided diffusion.

[Cuturi et al. 2022] Cuturi, M.; Meng-Papaxanthos, L.; Tian, Y.; Bunne, C.; Davis, G.; and Teboul, O. 2022. Optimal transport tools (ott): A jax toolbox for all things wasserstein. *arXiv preprint arXiv:2201.12324*.

[Cuturi 2013] Cuturi, M. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems* 26:2292–2300.

[Danskin 1966] Danskin, J. M. 1966. The theory of max-min, with applications. *SIAM Journal on Applied Mathematics* 14(4):641–664.

[Dhariwal and Nichol 2021] Dhariwal, P., and Nichol, A. 2021. Diffusion models beat gans on image synthesis. *arXiv preprint arXiv:2105.05233*.

[Esser, Rombach, and Ommer 2021] Esser, P.; Rombach, R.; and Ommer, B. 2021. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12873–12883.

[Ho, Jain, and Abbeel 2020] Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*.

[Jin et al. 2017] Jin, Y.; Zhang, J.; Li, M.; Tian, Y.; Zhu, H.; and Fang, Z. 2017. Towards the automatic anime characters creation with generative adversarial networks. *arXiv preprint arXiv:1708.05509*.

[Kantorovich 1942] Kantorovich, L. 1942. On the transfer of masses (in russian). In *Doklady Akademii Nauk*, 227–229.

[Karras et al. 2020] Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; and Aila, T. 2020. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8110–8119.

[Karras et al. 2021] Karras, T.; Aittala, M.; Laine, S.; Härkönen, E.; Hellsten, J.; Lehtinen, J.; and Aila, T. 2021. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems* 34.

[Karras, Laine, and Aila 2019] Karras, T.; Laine, S.; and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4401–4410.

[Kim and Ye 2021] Kim, G., and Ye, J. C. 2021. Diffusionclip: Text-guided image manipulation using diffusion models. *arXiv preprint arXiv:2110.02711*.

[Krantz and Parks 2002] Krantz, S. G., and Parks, H. R. 2002. *The implicit function theorem: history, theory, and applications*. Springer Science & Business Media.

[Li et al. 2017] Li, A.; Jabri, A.; Joulin, A.; and van der Maaten, L. 2017. Learning visual n-grams from web data. In *Proceedings of the IEEE International Conference on Computer Vision*, 4183–4192.

[Li, Jin, and Zhu 2021] Li, M.; Jin, Y.; and Zhu, H. 2021. Surrogate gradient field for latent space manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6529–6538.

[Murdock] Murdock, R. @advadnoun. https://twitter.com/advadnoun.

[Murdock 2021a] Murdock, R. 2021a. Big sleep: A simple command line tool for text to image generation, using openai's clip and a biggan.

[Murdock 2021b] Murdock, R. 2021b. Deep daze: A simple command line tool for text to image generation using openai's clip and siren (implicit neural representation network).

[Nichol and Dhariwal 2021] Nichol, A., and Dhariwal, P. 2021. Improved denoising diffusion probabilistic models. *arXiv preprint arXiv:2102.09672*.

[Nichol et al. 2021] Nichol, A.; Dhariwal, P.; Ramesh, A.; Shyam, P.; Mishkin, P.; McGrew, B.; Sutskever, I.; and Chen, M. 2021. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*.

[Patashnik et al. 2021] Patashnik, O.; Wu, Z.; Shechtman, E.; Cohen-Or, D.; and Lischinski, D. 2021. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2085–2094.

[Peyré and Cuturi 2019] Peyré, G., and Cuturi, M. 2019. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning* 11(5-6):355–607.

[Radford et al. 2021] Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*.

[Raffel et al. 2019] Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

[Séjourné et al. 2019] Séjourné, T.; Feydy, J.; Vialard, F.-X.; Trouvé, A.; and Peyré, G. 2019. Sinkhorn divergences for unbalanced optimal transport. *arXiv preprint arXiv:1910.12958*.

[Shorten and Khoshgoftaar 2019] Shorten, C., and Khoshgoftaar, T. M. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data* 6(1):1–48.

[Snell 2021] Snell, C. 2021. Alien dreams: An emerging art scene. https://ml.berkeley.edu/blog/posts/clip-art/.

[Song, Meng, and Ermon 2020] Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

[Thomee et al. 2016] Thomee, B.; Shamma, D. A.; Friedland, G.; Elizalde, B.; Ni, K.; Poland, D.; Borth, D.; and Li, L.-J. 2016. Yfcc100m: The new data in multimedia research. *Communications of the ACM* 59(2):64–73.

[van den Oord, Vinyals, and Kavukcuoglu 2017] van den Oord, A.; Vinyals, O.; and Kavukcuoglu, K. 2017. Neural discrete representation learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6309–6318.

7. Social aspects and evaluation

When happy accidents spark creativity: Bringing collaborative speculation to life with generative AI

Ziv Epstein[1*], Hope Schroeder[1*], Dava Newman[1,2]
[1]MIT Media Lab
[2]Human Systems Laboratory, Department of Aeronautics and Astronautics, MIT
{zive, hopes, dnewman}@mit.edu
[*]These authors contributed equally

Abstract

Generative AI techniques like those that synthesize images from text (text-to-image models) offer new possibilities for creatively imagining new ideas. We investigate the capabilities of these models to help communities engage in conversations about their collective future. In particular, we design and deploy a facilitated experience where participants collaboratively speculate on utopias they want to see, and then produce AI-generated imagery from those speculations. In a series of in-depth user interviews, we invite participants to reflect on the generated images and refine their visions for the future. We synthesize findings with a bespoke community zine on the experience. We observe that participants often generated ideas for implementing their vision and drew new lateral considerations as a result of viewing the generated images. Critically, we find that the unexpected difference between the participant's imagined output and the generated image is what facilitated new insight for the participant. We hope our experimental model for co-creation, computational creativity, and community reflection inspires the use of generative models to help communities and organizations envision better futures.

Introduction

New methods in generative machine learning, such as GANs, have created an explosion of opportunities and possibilities for computational creativity. One possibility GANs afford is the empowerment of people in casual creation (Compton and Mateas, 2015; Epstein et al., 2020a; Berns and Colton, 2020). For such casual creation, text-to-image models built on technologies like CLIP (Radford et al., 2021) have enormous promise by providing people an intuitive and user-friendly possibility space to query GAN-generated images via prompts (see Colton et al. (2021) for an overview). With such a technology, what are the possibilities for human creativity, and how might these applications impact communities? One opportunity for these text-to-image models is to aid in *imaginative idea visualization* (Colton et al., 2021) as a way to bootstrap the creative process. An issue of particular relevance to this goal is the inherent variety/fidelity trade-off of generative models: as the outputs of models become more realistic, they become less diverse (Ramesh et al., 2022). Yet it remains unclear how this trade-off impacts downstream tasks like imaginative idea visualization. On one hand, high-fidelity outputs might provide helpful details for how a given idea might be actually implemented. On the other, "happy accidents" from diverse but low-fidelity outputs might help evolve an idea via lateral thinking.

To explore how these two possibilities trade off, we design, deploy, and evaluate a computational creativity system for imagining and visualizing new ideas. Participants collaboratively speculated on utopian ideas for the future. These speculations were then fed as text prompts into a generative AI model to visually manifest them. We conducted a series of user interviews to learn about the experience from participants, and surface key themes. In this paper, we present a field report of the experience and use the system to trace broader questions about the social and collaborative aspects of creativity, such as when generative visual imagery can inspire ideas about the future, and how the variety/fidelity trade-off in generative models might impact creativity.

This work builds on the tradition of speculative design and design fiction, which use design to imagine and prototype alternative worlds (Dunne and Raby, 2013). The social nature of the collaborative speculation allows individuals to build these alternative worlds together. Furthermore, it explores how organizations or communities could use speculative design and design fiction to chart a course for the future.

There are several key contributions in this field report. First, we introduce a novel approach to prompt engineering, which leverages collaborative speculation from a facilitated co-design experience, rather than a single individual sourcing the prompts. Second, we highlight the importance of high-variance, low-fidelity images in inducing creative insights. Finally, we highlight the potential for computational creativity to aid in community dialogue and the collective elicitation of an organization's values. We explore this possibility by creating a zine to document and synthesize the findings, which we then present back to the community.

Methods

On October 8th, 2021, we organized a facilitated co-design experience at a solarpunk[1]-themed event at the MIT Media

[1] Solarpunk is an artform and aesthetic that imagines near-distant futures where humans have become climate change-

Lab, which had over 400 RSVPs. Participants in the exercise paired up and responded to the following prompt: “How will we re-imagine the following categories in utopia? Team up with someone, and each pick one of the following sectors: (see Table 1). Brainstorm how the future could be better at their intersection! Pick up a leaf and write your vision down on it. Add any visual representation you want. Tape the leaves to a stalk, add a flower, and put them in the solarpunk garden together!”

Money	Medicine	Cities	Transportation
Space	Agriculture	Music	Environment
Economy	Relationships	Family	Healthcare
Arts	Civil rights	Fashion	Infrastructure
Trade	Social justice	Oceans	Natural land & Wildlife
Education	Government	Energy	Community

Table 1: The 20 sectors used for prompt generation. Pairs of participants picked two to blend.

After co-creating their visions, participants placed their flowers with the vision written on it in the solarpunk garden (Figure 1). A total of 32 visions of the future were co-created by over 64 participants.

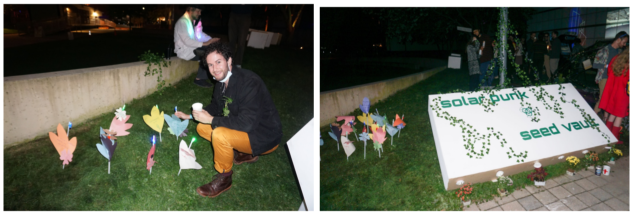


Figure 1: Solarpunk garden where participants placed their written visions. The garden was conceptualized as a “seed vault,” a place for hopeful seeds for the future to be stored and preserved.

Next, we took the 32 visions, and ran them through VQGAN+CLIP, a common model of text-to-image synthesis. This produced visual representations of the participants’ visions, and served as the output of the speculation experience. We then used these images for evaluation of the paradigm and incorporated them into a zine for additional community impact.

Evaluation via user interviews To evaluate the experience of facilitated speculation augmented with generative AI, we conducted a series of ten 15-20 minute semi-structured interviews from March 30th to April 11th, 2022 about 10 unique visions both in person and by video chat.

First, to measure relatedness between prompt and image, we tested if participants could recognize the image generated by their prompt. For each participant, we paired the image that corresponded to their prompt with three additional images from other prompts and presented the set of four to the participant in random order. We invited participants to identify the image generated from the prompt they wrote in the original activity. Regardless of their answer, we then revealed the correct image. We then asked questions related to their interpretation of the image, starting by asking them to describe what they saw. Based on directions from this initial description, we then asked follow-up questions about if and how the image differed from their expectations. Finally, we asked if the participant had ideas for a follow-up prompt that better reflected their vision. After interviews were complete, we coded them using standard qualitative coding practices for themes that emerged across conversations (Saldaña, 2013) .

Results

While all loosely related to the solarpunk theme, the written prompts from participants in the original activity had a remarkable amount of diversity. Some participants generated futuristic ideas, while others called for a return to traditional wisdom or practices. Four exemplary images, with their corresponding prompts are shown in Figure 2.

User Interviews We conducted 10 interviews with individuals who participated in the original activity. We focused our interviews to participants who remembered the conversation they had from several months prior and still recalled the vision that they wrote down.

We coded our 10 participant interviews for major themes distilled through repeated interviewing. In the process of synthesizing recurring themes across conversations, we noticed that many participants gleaned new insight as a result of viewing the image. We recorded when participants mentioned new information gleaned about their original idea after viewing the generated image. A majority of participants reported gaining new ideas as a result of viewing the image generated by their prompt, and this new information fell into two main categories.

The first was the emergence of new, unexpected ideas for implementing the vision they had written (40% of interviews, n = 4). For example, the creator of the prompt “Biophilic vertical gardens lining neighborhood roads, creating function and beautiful public spaces” already had a “pretty concrete” image of what their vision could look like given that biophilic community gardens already exist. However, they had imagined “pumps and tubes” as a visual feature given the complex engineering required to create such gardens. When viewing the generated image, the creator noted that the wall looked like it was made of a “natural, rocky substrate” rather than one with exposed hydroponic engineering (see Figure 2, left). This lead the participant to note that perhaps the wall of the garden could in fact be beautiful and natural-looking as well as functional, and that this aesthetic would be an improvement over exposed pipes.

The second main area of insight participants reported in interviews related to generating unexpected connections between a vision and lateral concepts (40%, n = 4). For example, the prompt “Public spaces: solidarity-building. The intersection of oceans and relationships. Publicly accessible oceanic vistas” generated an image of humans on a beach (see Figure 2, center left). Something like sand is present,

resilient and learned to live in harmony with nature.

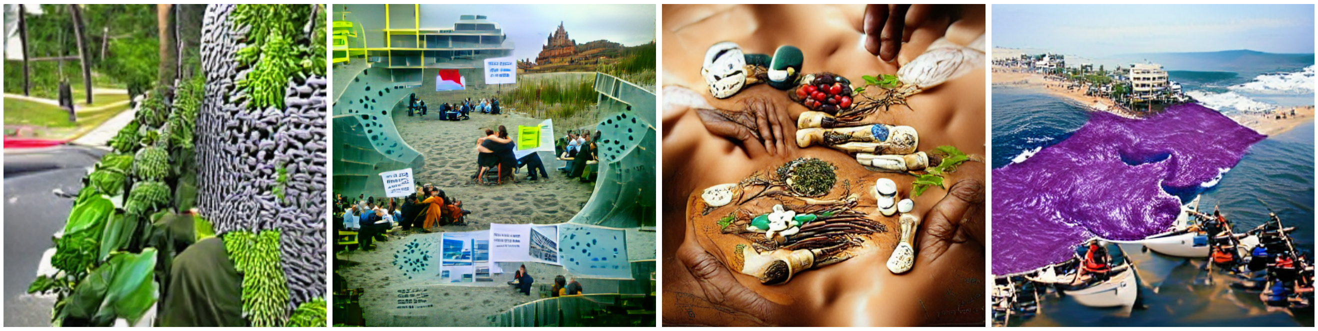


Figure 2: Four images from the following prompts: Biophilic vertical gardens lining neighborhood roads, creating function and beautiful public spaces (left), Public spaces: solidarity-building. The intersection of oceans and relationships. Publicly accessible oceanic vistas (center left), Holistic traditional medicine as an art form (center right), Dye the ocean purple to prevent global warming (right)

but the lack of an actual ocean in the image was surprising to the prompt's creator. However, the repeated visual motif of people in relationship yielded a new perspective for the creator, who then reflected on the importance of relationship for organizing in public space. This was not a framing he had not been considering as central to this prompt before being presented with the image. The lack of ocean in an image generated from a prompt with two references to the ocean could be considered low-fidelity and ultimately undesirable behavior from VQGAN+CLIP. Yet the unexpectedness of the image led the prompt's creator in a new direction that was ultimately valuable for ideation. As a result, we consider this a happy accident in the context of this exercise in social dreaming.

In another example, the participant who wrote "Holistic traditional medicine as an art form" reported noticing the centrality of hands as a visual motif in the image (see Figure 2, center right). Her interpretation was that the hands made the image "focused on the making process" and that the image "emphasizes labor." She reported that the human labor aspect behind traditional medicine was not a major association she had with the topic before viewing the image.

Most (80%, n = 8) interviewees mentioned at least one idea for a follow-up prompt to refine the image or clarify their vision. For example, the biophilic garden image featured a visual element on the bottom left that looked like a road. The prompt's creator noticed the curb and said they might use language like "neighborhood path" instead of "road" to generate an image with explicitly pedestrian streets in the future.

Despite the fact that it was easy for participants to identify the image created from their prompt (90%, n = 9) out of a set of random images, most participants interpreted the generated images as being either partly or substantially different from what they imagined prior to seeing the image (70%, n = 7). Three interviewees that did not gain any new insight from viewing the images were concerned that the image was too abstract to be useful. On the other hand, two other participants commented that they specifically appreciated the "whimsical" image of a technical concept.

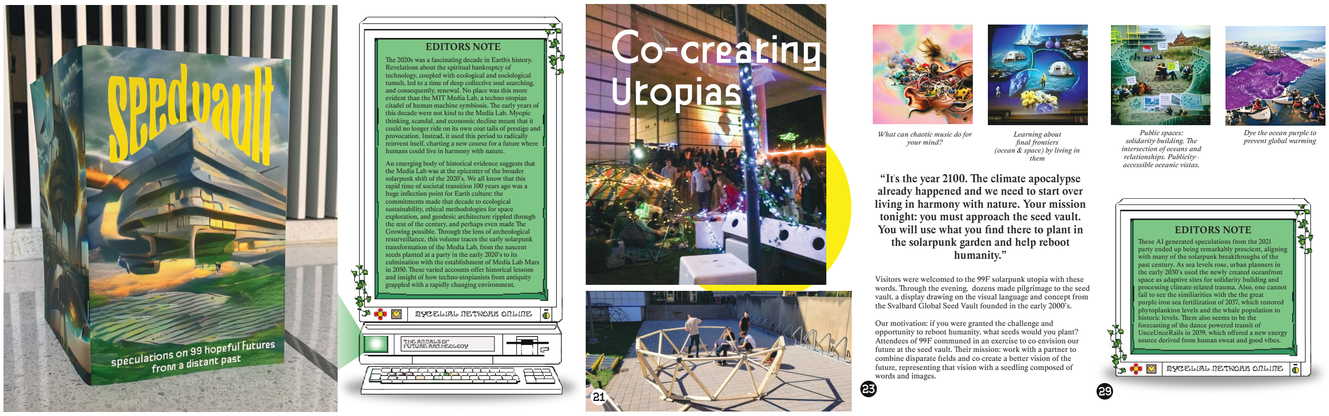
"When the place inspires the zine inspires the place"[2] In the wake of the solarpunk event, we created a zine outlining the project and showcasing all the co-written visions and corresponding GAN images to aid in community reflection (see Figure 3). We printed over 500 copies and distributed them to community stakeholders. As community members, we wanted to give these images back to the community, hoping they would act as a mirror for additional conversation and "close the loop" of the reflective process.

The event, co-creation experience, and zine all came at a time when our organization was emerging from the throes of the pandemic and actively setting its strategy and vision for the next decade and beyond. At such a critical inflection point, the combination of these three elements had several key cultural impacts. First, there was the physical aspect of people coming together to imagine and plant the seeds for the future after a long period of pandemic-induced isolation. Second, in an organization with varied interests and priorities, the process of brainstorming through images helped visualize and synthesize collective threads across the organization. Finally, cementing the images in familiar, tangible form of printed media provided a snapshot for both disseminating the outputs and orienting discussion around the organization's future (Pérez y Pérez and Ackerman, 2020). The zine has since been used to communicate interests and priorities to both visitors and external stakeholders.

Discussion

Initially, we predicted a main contribution of this work would be designing a tool to concretize visions that teams co-created. We found that occasionally the image did meaningfully crystallize a vision for the future in a literal way, and many participants noted the image gave them ideas for ways to implement their vision. However, we often found that it was the unexpected differences between the prompt and the generated image's interpretation of it that yielded new insight for and excitement from participants. This suggests that it was the high-variance, low-fidelity behavior of

[2] Quote by Sidebody `https://www.instagram.com/p/CbtJnWlloui/`


Seedvault
speculations on 99 hopeful futures from a distant past
MYCELIAL NETWORK ONLINE
THE ANNALS OF FUTURE ARCHEOLOGY
Co-creating Utopias
21
23
29

EDITORS NOTE

The 2020s was a fascinating decade in Earth's history. Revelations about the spiritual bankruptcy of technology, coupled with ecological and sociological tumult, led to a time of deep collective soul searching, and consequently, renewal. No place was this more evident than the MIT Media Lab, a techno-utopian citadel of human-machine symbiosis. The early years of this decade were not kind to the Media Lab. Myopic thinking, scandal, and economic decline meant that it could no longer ride on its own coat tails of prestige and provocation. Instead, it used this period to radically reinvent itself, charting a new course for a future where humans could live in harmony with nature.

An emerging body of historical evidence suggests that the Media Lab was at the epicenter of the broader solarpunk shift of the 2020's. We all know that this rapid time of societal transition 100 years ago was a huge inflection point for Earth culture: the commitments made that decade to ecological sustainability, ethical methodologies for space exploration, and geodesic architecture rippled through the rest of the century, and perhaps even made The Growing possible. Through the lens of archeological resurveillance, this volume traces the early solarpunk transformation of the Media Lab, from the nascent seeds planted at a party in the early 2020's to its culmination with the establishment of Media Lab Mars in 2050. These varied accounts offer historical lessons and insight of how techno-utopianists from antiquity grappled with a rapidly changing environment.

What can chaotic music do for your mind?

Learning about final frontiers (ocean & space) by living in them

Public spaces: solidarity-building. The intersection of oceans and relationships. Publicity--accessible oceanic vistas.

Dye the ocean purple to prevent global warming

"It's the year 2100. The climate apocalypse already happened and we need to start over living in harmony with nature. Your mission tonight: you must approach the seed vault. You will use what you find there to plant in the solarpunk garden and help reboot humanity."

Visitors were welcomed to the 99F solarpunk utopia with these words. Through the evening, dozens made pilgrimage to the seed vault, a display drawing on the visual language and concept from the Svalbard Global Seed Vault founded in the early 2000's.

Our motivation: if you were granted the challenge and opportunity to reboot humanity, what seeds would you plant? Attendees of 99F communed in an exercise to co-envision our future at the seed vault. Their mission: work with a partner to combine disparate fields and co-create a better vision of the future, representing that vision with a seedling composed of words and images.

EDITORS NOTE

These AI-generated speculations from the 2021 party ended up being remarkably prescient, aligning with many of the solarpunk breakthroughs of the past century. As sea levels rose, urban planners in the early 2030's used the newly created oceanfront space as adaptive sites for solidarity building and processing climate-related trauma. Also, one cannot fail to see the similiarities with the the great purple-iron sea fertilization of 2037, which restored phytoplankton levels and the whale population to historic levels. There also seems to be the forecasting of the dance powered transit of UnceUnceRails in 2039, which offered a new energy source derived from human sweat and good vibes.

Figure 3: Cover and four select pages from the Seedvault zine. Written in the style of speculative fiction, the zine takes place in the year 2121, recounting an event that took place 100 years ago and how the previous 100 years unfolded from there.

VQGAN+CLIP that yielded these unexpected and whimsical differences, which in turn induced novel and creative insights. This finding has important implications for the design of image synthesis systems for computational creativity. As new systems become increasingly realistic (such as new models like DALL-E 2 (Ramesh et al., 2022)), there is a danger that this increased fidelity will come at the cost of these unexpected quirks that we found actually sparked positive lateral thinking in our participants.

By offering a low-friction and provocative new way to integrate visual storytelling with community engagement, text-to-image models with collaboratively sourced prompts have an opportunity for cultural and organizational impact. This approach could be used in contexts as diverse as crafting organization mission statements, co-designing community interventions, and facilitating a mediation process.

There are, however, dangers to such an approach. For one, AI-generated images are bounded by training data, which inherits historical biases and cultural practices (Ganimals Blog, 2020; Crawford and Paglen, 2019). Therefore uncritically relying on model outputs as an "oracle" may entrench users in existing inequities and stereotypes, rather than freeing them to envision radical new possibilities. Relatedly, if such an approach becomes commonplace, there is the risk that such visualization strategies could be a crutch if used too much, with users becoming overly reliant on a machine's vision of the future rather than their own. Finally, there is the risk that anthropomorphizing the AI can undermine human credit and responsibility (Epstein et al., 2020b).

It is also important to reflect on the notion of "utopias," which we used to frame this collaborative speculation. Literally meaning "no place" in the original Greek, utopias represent idealized non-existent or impossible societies. Rarely in utopian thinking are questions like "A utopia for whom?" or "A utopia at what cost?" considered, which in turn leads to imagined futures that are culturally homogeneous and can perpetuate existing inequities. Furthermore, the very concept of utopias imagines a future society different from and evolved upon the current one, a notion rooted in endless growth, a potentially colonial and capitalistic value (Morrison, 2017). Rather than envisioning radical new alternatives to be manifested, many indigenous cultures instead hold a worldview that humans are a part of a complex web of ecological relationships that can exist an a perpetual steady-state equilibrium (Kimmerer, 2013; Sepie, 2017). In the prompts we received, we saw varied interpretations of and priorities for utopias: some participants longed for a return to local economies or traditional medicine, while others dreamt of futuristic space suits and wind turbines. Finally, it is important to note that utopias are but one frame for imaginative idea visualization. While we used the idea of utopia to foster brainstorming about the future, we believe the paradigm could instead focus inward for any community or space looking to collaboratively speculate about its future.

Our work has several limitations which open up exciting possibilities for future work. For one, we focused on one particular community – the MIT Media Lab, a technology-focused research institution embedded in a university. Future work could explore how this approach works for other communities with distinct cultures and practices, and how the effectiveness of the approach varies across community contexts. In addition, we relied on analog media throughout the process: from handwritten prompts by participants and a paper zine distributed in person to us showing participants generated images asynchronously in the evaluation phase. Future work could explore digital versions of these methods, such as an online interface for collaborative prompt generation, an online gallery for the prompts and their corresponding images, or real-time dialogue with a generative model. We also did not include in-depth explanation for how the generative model worked. Future work should consider more explicitly users' understanding of the involved technology, as well as consider other types of models, such as physically-informed models of climate futures (Lütjens et al., 2021b,a) or those that integrate across larger, more complex systems (Lavin et al., 2021). Finally, there is the possibility of using this method in diverse settings. We hope that this approach could be applied to other community contexts, whether it be designs for a local community garden or bold new tactics to fight the global climate emergency.

Acknowledgements

We thank Eyal Perry, Micah Epstein, Amy Smith, Simon Colton, Rubez Chong, Océane Boulais, Maggie Hughes, Valdemar Danry and Tobin South for helpful feedback, comments and support. We thank all participants for their time.

Author Contributions

ZE, HS, and DN conceptualized the project. HS conducted the user interviews. ZE and HS wrote the paper, with input from DN.

References

Berns, S., and Colton, S. 2020. Bridging generative deep learning and computational creativity. In *ICCC*, 406–409.

Colton, S.; Smith, A.; Berns, S.; Murdock, R.; and Cook, M. 2021. Generative search engines: Initial experiments. In *ICCC*.

Compton, K., and Mateas, M. 2015. Casual creators. In *ICCC*, 228–235.

Crawford, K., and Paglen, T. 2019. Excavating ai: The politics of images in machine learning training sets. *AI and Society*.

Dunne, A., and Raby, F. 2013. *Speculative everything: design, fiction, and social dreaming*. MIT press.

Epstein, Z.; Boulais, O.; Gordon, S.; and Groh, M. 2020a. Interpolating gans to scaffold autotelic creativity. *International Conference on Computational Creativity Causal Creator Workshop*.

Epstein, Z.; Levine, S.; Rand, D. G.; and Rahwan, I. 2020b. Who gets credit for ai-generated art? *Iscience* 23(9):101515.

Ganimals Blog. 2020. Beware the training data: The barracuda effect. ganimals.media.mit.edu.

Kimmerer, R. 2013. *Braiding sweetgrass: Indigenous wisdom, scientific knowledge and the teachings of plants*. Milkweed editions.

Lavin, A.; Gilligan-Lee, C. M.; Visnjic, A.; Ganju, S.; Newman, D.; Ganguly, S.; Lange, D.; Baydin, A. G.; Sharma, A.; Gibson, A.; et al. 2021. Technology readiness levels for machine learning systems. *arXiv preprint arXiv:2101.03989*.

Lütjens, B.; Crawford, C. H.; Veillette, M.; and Newman, D. 2021a. Pce-pinns: Physics-informed neural networks for uncertainty propagation in ocean modeling. *arXiv preprint arXiv:2105.02939*.

Lütjens, B.; Leshchinskiy, B.; Requena-Mesa, C.; Chishtie, F.; Díaz-Rodríguez, N.; Boulais, O.; Sankaranarayanan, A.; Piña, A.; Gal, Y.; Raïssi, C.; et al. 2021b. Physically-consistent generative adversarial networks for coastal flood visualization. *arXiv preprint arXiv:2104.04785*.

Morrison, M. I. 2017. *Decolonizing Utopia: Indigenous Knowledge and Dystopian Speculative Fiction*. University of California, Riverside.

Pérez y Pérez, R., and Ackerman, M. 2020. Towards a methodology for field work in computational creativity. *New Generation Computing* 38(4):713–737.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 8748–8763. PMLR.

Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*.

Saldaña, J. 2013. *The coding manual for qualitative researchers*. SAGE.

Sepie, A. J. 2017. More than stories, more than myths: Animal/human/nature (s) in traditional ecological worldviews. *Humanities* 6(4):78.

A Data-Driven Architecture for Social Behavior in Creator Networks

Berkeley Andrus, Nancy Fulda
Computer Science Department
Brigham Young University
Provo, UT 84602 USA
berkeleyandrus@gmail.com, nfulda@cs.byu.edu

Abstract

Previous Computational Social Creativity work has improved the performance of automated creators using social mechanics inspired by human behavior. However, these simulations have often focused on generic or assumed human behaviors rather than on specific anthropoligical data. In this work we take a more focused approach by comparing simulated social behavior to observed behavior in large social networks of human creators. We analyze social patterns among human creators by defining metrics for social behavior within creative communities and collecting data for three online communities of creators: Scratch, FanFiction, and r/ArtCrit. We introduce the Architecture for Multi-Agent Creative Societies (AMACS), a modeling tool which controls the social activity of automated creators and can be adapted to any creative discipline. We demonstrate AMACS's ability to recreate a wide range of network-level social behaviors, including the behaviors observed in three human societies.

Introduction

Social interaction has long been understood to be an essential component of the creative process (Csikszentmihalyi 2014; Boden 1992; Glăveanu 2013; Jennings 2010). Social interactions help creators in many disciplines by facilitating encouragement, correction, inspiration, and mentorship. A creator's social circles provide opportunities to test out new ideas, collaborate, and hone skills. This is true in disciplines ranging from pottery to programming to dance.

Past research on the role of social interaction in the creative process has typically taken one of two approaches. The first approach has been to analyze social networks of creators directly and identify quantitative and qualitative trends relevant to specific facets of creativity (Sylvan 2007; Sylvan 2010; Xu and Bailey 2012; Crain and Bailey 2017; Marlow and Dabbish 2014; Campbell et al. 2016; Evans et al. 2017; Pace et al. 2013). The second approach has been to simulate networks of creators in fully automated social environments (Linkola and Hantula 2018; Hantula and Linkola 2018; Gómez de Silva Garza and Gero 2010; Greenfield and Machado 2009; Alnajjar and Hämäläinen 2018; Hämäläinen, Alnajjar, and others 2019). These simulations tend to focus on generic or assumed rules of human behavior rather than on quantitative data, and while they have the potential to inform our understanding of human creativity (Saunders and Bown 2015) they are often more concerned with improving the performance of simulated creators. Surprisingly, these approaches have rarely - if ever - been mixed. Researchers have attempted to *observe* or *simulate* the social behavior of creators, but not both.

In this work we combine these two approaches of measuring and simulating user behavior. To our knowledge it is the first attempt to quantitatively measure and then replicate the social behaviors of creators acting in a social network. This data-driven and focused approach allows for more meaningful analysis of simulated creators, making simulation a viable tool for understanding human creativity and potentially improving automated creative social systems.

We also introduce an Architecture for Multi-Agent Creative Societies (AMACS), a simulation architecture implemented in Python that controls the social activity of automated creators and can be adapted to any creative discipline. We make AMACS publicly available and hope it will act as a common test bed for future researchers experimenting with social mechanics for automated creative agents.

Related Work

Many researchers, both in the field of psychology and in artificial intelligence, have sought to define and understand the role of social interaction in creativity. For example, Csikszentmihalyi (2014) argued that creativity is only possible when a creator interacts with a domain of cultural knowledge and a field of peers, making creativity an inherently social process. Boden (1992) discussed creativity in terms of conceptual spaces, which she defined as being "familiar to (and valued by) a certain social group" rather than belonging solely to an individual. Jennings (2010) proposed using socialization as a tool for increasing the autonomy of simulated creators. Glăveanu's framework of creativity (2013) elevated the importance of socialization in creativity by including *audience* as a key member of the creative process.

Parallel to the effort to define the social aspects of creativity has been the effort to quantitatively observe them, specifically in online social environments. Sylvan (2007; 2010) used the term 'Online Community of Creators' (OCOCs) to describe social network sites where creators share and receive feedback on their work. She selected two OCOCs - The Village and Scratch - and attempted to track how ideas spread through these online communities by finding qualities correlated with influential individuals and artifacts. Xu and Bailey (2012) analyzed interactions between users in the online photography critique community PhotoSIG, focusing specifically on critique mechanisms. Crain and Bailey (2017) analyzed how users engage with criticism on three art critique subreddits, focusing on the quality of feedback and how it impacted a creator's willingness to iterate on published artifacts. Marlow and Dabbish (2014) investigated how users of Dribble gradually become more skilled at their craft. Campbell and associates (Campbell et al. 2016; Evans et al. 2017) also explored how OCOCs allow creators to improve, framing their findings with a model they call *distributed mentoring*. Pace et. al. (2013) mapped theories concerning more traditional (i.e. offline) creative communities to OCOCs while analyzing the role of leaders in the Etsy community.

There has also been much work done to simulate the social behavior of creators, a task which Saunders and Brown (2015) describe as 'Computational Social Creativity'. Hantula and Linkola (Linkola and Hantula 2018; Hantula and Linkola 2018) study *collaborator selection* in a simulated society of image-generating agents with various changing tastes. Gómez de Silva Garza and Gero (2010) introduce a network in which agents are engaged in either creating or evaluating simple visual designs. Greenfield and Machado (2009) use the same distinction between agents, calling their agents 'artists' and 'critics'. Critics in their system compare agent-generated artifacts to human-generated ones via representative vectors. Alnajjar and Hämäläinen (2018; 2019) use a social network which contains only a master and an apprentice. The master generates training data for the apprentice and curates a dataset of human-generated examples for the apprentice to learn from.

There are examples in which multiple simulated agents work together to generate a single artifact (Pérez y Pérez et al. 2010; Boyd, Hushlak, and Jacob 2004; Wright, Purver, and others 2020). Because these social networks are focused only on collaborating (rather than sharing and evaluating finished artifacts, forming relationships, etc.) they fall outside the scope of the creative societies we are interested in here.

To our knowledge, the present work is the first attempt to both observe and simulate the social interactions between creators, an important bridge between these previously disjointed approaches. It also introduces the first discipline-agnostic simulation tool for creative societies of which we are aware. Our hope is that this combined approach and the accompanying software package will add more meaning and focus to future approaches at social simulation.

Analyzing Creative Societies

The purpose of this work is to create a data-driven simulation of the social behaviors of creators. In order to validate that simulations are acting in a human-like manner, we need a framework for analyzing and describing both human and automated societies so that different societies can be meaningfully compared with one another.

To this end we introduce a quantitative analysis framework that consists of four metrics: Creator to Agent Ratio, Reciprocity, Clustering, and Attention Concentration (each defined below). These metrics were chosen because they each affect the experience of individual agents and can be calculated based on publicly available information as described below.

Creator to Agent Ratio (CAR) is the percentage of community members that create original artifacts (paintings, songs, programs, etc.) as opposed to only commenting on the artifacts of others. CAR is defined as $\frac{|C|}{|A|} \times 100$, where C is the set of all creators in the network and A is the set of all agents (both creators and non-creators) in the network.

A network's CAR is important in defining the relationship between creators and fans. A high CAR can make it difficult for creators to build audiences because the have more competition, while a low CAR might make it difficult for fans to find creators they like.

Reciprocity is the tendency for an agent to return the favor when another agent gives feedback on one of their artifacts. In an online setting, feedback can include comments, 'Likes', or any other publicly observable recognition of the artifact. We define reciprocity as $P(A \triangleright B \mid B \triangleright A) \times 100$, where A and B are distinct creators in the network and $X \triangleright Y$ denotes that an agent X has provided feedback for an artifact generated by agent Y at some point in the past.

Reciprocity describes one way in which network agents form relationships with one another. If agents are inclined to reciprocate positive attention or feedback, then it becomes easier for a relationship to form out of a single agent's desire for a connection. High reciprocity also means that a network rewards good behavior through reciprocation, which incentivizes agents to be generous with one another.

Clustering is the tendency for an agent to be friends with its own friends-of-friends. Highly clustered networks indicate the presence of cliques and sub-communities within the larger network. The definition of clustering depends on a definition of 'friendship', which takes different forms in different types of communities. For consistency, we notate that two distinct agents A and B are friends using $A \diamond B$ and say that $A \diamond B \iff (A \triangleright B) \cap (B \triangleright A)$. In other words, if two creators have commented on each others artifacts at least once each, we call them friends.

Given a definition of friendship, we define clustering as $P(B \diamond C \mid A \diamond B, A \diamond C) \times 100$ for any three distinct creators A, B, and C in the network. This is equivalent to the global clustering coefficient for graphs if we consider each agent as a node and each friendship as an undirected edge.

A network's clustering rate can serve as an indicator for how opinions and ideas spread through a population of agents (Malik and Mucha 2013; Centola 2010; Jackson 2019). Tight clusters can cause agents to become more similar to their direct contacts, but they can also insulate agents and slow the spread of globally popular beliefs (Granovetter 1973).

Attention Concentration measures how popular the most popular artifacts in the network are, where popularity is defined as the volume of comments received. We measure attention concentration using the Gini coefficient (Gini 1912), a metric commonly used to describe the wealth in-

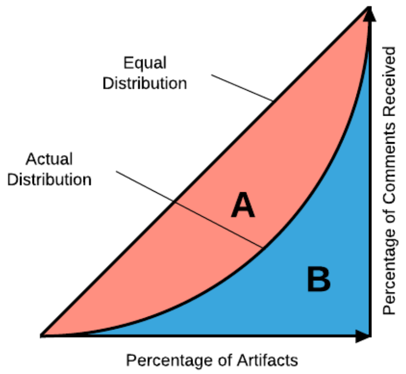


Figure 1: Visualization of the Gini Coefficient, used to measure the concentration of attention within a network of creators. The curved line shows the percentage of all comments received by the corresponding percentage of artifacts, which are sorted by ascending popularity. The diagonal line shows a hypothetical society where all artifacts receive an equal number of comments. The Gini Coefficient is the area of A divided by the sum of the areas of A and B.

equality of a population. We refer to (Dorfman 1979) for a mathematical definition, but a basic explanation is provided in Figure 1. The Gini Coefficient can be understood as a real number in the range [0, 1] where 0 indicates that all artifacts receive an equal amount of feedback and 1 indicates that all feedback is directed at a single artifact.

Attention concentration can be a significant pain point for human creators, especially in online environments. Xu and Bailey (2012) found that over 80% of artifacts on a photography sharing community received fewer comments than average users considered useful, while other artifacts received many comments. If most attention is being directed at a small handful of popular artifacts, it can be difficult for new creators to feel engaged with the network.

Data Collection

In order to understand the social behavior of human communities, we apply this analysis framework to several communities of creators. Our purpose is to collect quantitative data that can then be used to validate simulations of social behavior. In this work we focus on online communities of creators that are large, include mechanisms for artistic critique in the form of comments, and permit legal scraping of user and artifact data. After considering nearly a dozen communities, we select three that best fit the above criteria: Scratch, FanFiction, and the r/ArtCrit community on Reddit. These communities are oriented towards programming projects, creative writing, and visual artwork respectively.

To collect data for Scratch, we use a Selenium-based web scraper to collect several thousand of the most recent artifacts published in the 'Music' category of coding projects. For each recent project we then find the user who created that project and collect data on each project published by that user. For each project we collect the project ID and the list of all users who have commented on that project. We collect a total of 91,506 projects and 82,952 comments posted by 39,631 users.

Table 1: Observed behavior in three online communities of creators using four network-level metrics.

Community	CAR	Reciprocity	Clustering	AC
Scratch	17.1	1.2	6.3	0.901
FanFiction	23.2	11.0	15.6	0.762
r/ArtCrit	59.6	0.5	1.0	0.489

Following (Milli and Bamman 2016), we scrape FanFiction data using Python-generated HTTP requests and parse responses with the BeautifulSoup library. We select 32 of the most popular book 'canons' (the original works that FanFiction stories are based on) and scrape all stories and comments related to those canons, excluding anonymous comments. We collect 189,076 stories and 7,789,744 comments posted by 387,253 users.

We access r/ArtCrit data using Cornell University's ConvoKit toolkit (Chang et al. 2020). The dataset includes 14,201 posts and 33,451 comments made by 11,992 users. We filter out posts or comments made by users who have since deleted their accounts (as these are effectively anonymous), comments that are responses to other comments rather than to posts, and comments made by the same user as the post being commented on.

Anonymized copies of the collected data are available upon request. In accordance with the privacy policies of Scratch and FanFiction, this anonymized data will include only the metadata necessary to calculate the four metrics described above, not user data or content of the posted artifacts or comments themselves. All scraping that we performed was in accordance with the respective site policies.

Human Analysis Results

The results of applying our framework to these three communities are found in Table 1. We note that there is a wide variance in the behavior of these three communities. For example, r/ArtCrit's CAR is more than double the other two communities' and FanFiction has a much higher Reciprocity and Clustering rate than the other two. Further analysis of these results are provided in (Andrus 2021), but for our purposes here we are primarily interested in recreating these quantitative behaviors in a simulated environment.

AMACS: the Architecture for Multi-Agent Creative Societies

Having observed several human creative societies, we are now prepared to simulate them. To this end we introduce AMACS: the Architecture for Multi-Agent Creative Societies. AMACS is a flexible, task-agnostic architecture implemented in Python that defines how automated agents generate and evaluate creative artifacts. It also defines how agents form relationships with and are influenced by one another. Any designer who desires to use AMACS to simulate the behavior of agents in a given creative discipline need only implement functions for evaluating and generating artifacts within that discipline; AMACS handles the rest, including content discovery, social mechanics, and the changing aesthetic tastes of agents. We hope that it will serve as a test bed and reference point for future researchers who wish to perform experiments in a common setting. The full AMACS implementation and three example instantiations are provided online at `https://github.com/bandrus5/amacs`.

AMACS Methodology

An AMACS network, similar to previous simulated creative societies (Hantula and Linkola 2018; Linkola and Hantula 2018; Gómez de Silva Garza and Gero 2010; Greenfield and Machado 2009), is composed of a pool of agents capable of generating and evaluating creative artifacts. Agent aesthetic tastes change over time as agents interact with and are influenced by one another. Unlike previous works, an AMACS network can be implemented for any creative task (e.g. writing poetry, designing furniture, or composing music), and it includes hyperparameters that can be tuned to elicit specific human-like behaviors.

Following Hantula and Linkola (Hantula and Linkola 2018), each agent in an AMACS network has individual aesthetic tastes represented by numeric scores. In Hantula and Linkola's simulations, which use image generation as the creative task, each agent's tastes are represented by a single number that corresponds to their preferred value along some evaluative spectrum such as Symmetry, Contrast, etc. We expand this evaluative paradigm with a multidimensional "artifact space". Unlike in Hantula and Linkola's simulations, the AMACS artifact space can have as many dimensions as needed, and all agents share the same artifact space. We consider the artifact space to be an application of Boden's *conceptual space* (Boden 1992), albeit a relatively simple one.

Each dimension of the artifact space corresponds to some evaluative function. The nature of these evaluation functions will depend on the creative task of the network. For example, in a music-generation AMACS network, dimensions of the artifact space might correspond to tempo, key, and sentiment. Dimensions can represent binary distinctions (e.g. whether or not a poem conforms to a 5-7-5 syllabic pattern) or real value measurements (e.g. the type-token ratio of a short story). They can even be unbounded (e.g. the length of a song), though in many cases there will be an inherent lower and upper limit (e.g. the percentage of image pixels that are blue cannot fall outside the range [0, 100]).

Points within the artifact space can be used to describe both artifacts and agent preferences. Each artifact is assigned a score vector S which situates that artifact within the artifact space. We represent an agent's preferences with a taste vector T and a taste weight vector W. T describes which point within the space the agent considers to be 'perfect', and W allows the agent to scale the artifact space and choose which dimensions it cares most about. S, T, and W each have the same dimensionality as the artifact space.

Some dimensions of the artifact space may have a "correct" answer, meaning that all agents share the same taste values in those dimensions. This allows an AMACS designer to enforce artistic constraints, such as that all poems must rhyme or that all songs must be in a major key. In the other dimensions agents are free to set their own tastes, giving them creative freedom to choose, for example, the key a song is written in or the dominant color used in a painting. The fact that agents are fixed in some dimensions and not in others allows for a shared understanding of which artifacts are valid but individual understanding of which artifacts are good. This is partially inspired by Wiggins's (2006) rule sets $\mathscr{R}$ and $\mathscr{T}$ for constraining and traversing a conceptual space respectively. In AMACS, agents and artifacts are constrained in membership dimensions (analogous to Wiggins's $\mathscr{R}$) and are free to traverse attribute dimensions (analogous to Wiggins's $\mathscr{T}$). Future work may attempt to simulate more transformative creativity by allowing agents to ignore membership dimensions of the artifact space under specific conditions or invent new attribute dimensions and add them to the global artifact space. This latter approach was described but not implemented by Ventura (2019).

On each network time step, each existing agent has an opportunity to generate a new artifact, analogous to a human creator sharing a new piece of artwork with their social network. The decision of whether to produce an artifact is based on whether the agent produced anything on the previous time step and the average value of W, which along with scaling the artifact space is used to model the agent's confidence in its own tastes.

Once all agents have had an opportunity to generate artifacts, each agent evaluates a small number of artifacts from the current or past time steps. Each agent is given a list of recommended artifacts, and the agent randomly samples from the recommendations based on its own criteria. It then evaluates each chosen artifact a using the following value definition:

$$value(a) = -\sum_{i=1}^{n} |\boldsymbol{S}_i - \boldsymbol{T}_i| * \boldsymbol{W}_i \tag{1}$$

where n is the number of dimensions in the artifact space and S is the vector representing a's location in the artifact space. This is equivalent to the negative weighted Manhattan distance between T and S.

The agent can then choose whether to leave publicly observable feedback for the artifact, which can take several different forms. The simplest form of feedback, which we use in our experiments, is a binary feedback system analogous to the 'Like' button found on many online platforms for creators. If an agent's evaluation of an artifact falls above some threshold, the agent gives the artifact a 'Like' and the evaluation is classified as favorable. A more ambitious feedback system could include agents leaving some sort of comment on the artifact with details about what they liked or didn't like. Alternatively, an agent could provide feedback on an artifact by editing it to something the evaluating agent likes better and sharing the 'enhanced' version of the artifact with the original creator. We leave these more complex mechanisms to future work.

The final process on each time step is for agents to adapt based on what they've experienced on the current time step. First, agents change their taste weights to reflect their changing confidence. Taste weights go up if agents received positive feedback from their peers and/or they evaluated artifacts that they liked. Otherwise taste weights go down. Next, agents have a small probability of changing their tastes. The higher their taste weights, the lower probability their tastes will change. If an agent chooses to change its tastes, it typically moves towards its most recently evaluated artifacts, although in some cases it will move away. Finally, agents choose whether or not they will generate an artifact on the next time step. All changes to tastes and taste weights happen independently in each dimension of the artifact space.

In our experiments we initialize each AMACS network with a pool of 42 agents with random tastes and taste weights. We run the network for 30 time steps, adding 4 new agents to the network on each time step to simulate the community growing over time. There are many other population mechanics that could be explored in future work depending on the specific human behavior being simulated.

We increase AMACS's flexibility by defining 11 hyperparameters that affect agent behavior, specifically in how they select new artifacts to interact with. Each hyperparameter loosely corresponds to design decision that community administrators control, which makes them useful for tuning AMACS networks to resemble specific communities. They are:

- **Agent Taste** controls how personalized the recommendations made to AMACS agents are. This is analogous to the level of customization on websites used by human creators to find new content.
- **Creator Familiarity** controls how much an AMACS agent prefers to review artifacts created by other agents is has interacted with in the past, regardless of whether those interactions were positive or negative. **Creator Favorability** is similar, but it includes only positive interactions.
- **Mutual Contact** controls an agent's preference to review artifacts created by other agents who share a mutual contact. **Mutual Friend** is similar, but it includes only contacts where most interactions have been positive.
- **New Artifact** controls the extent to which AMACS promotes artifacts generated on the most recent time steps.
- **Popular Artifact** controls whether AMACS promotes artifacts based on their number of positive reviews.
- **New Creator** controls the extent to which AMACS promotes artifacts generated by agents who have not generated many artifacts in the past.
- **Popular Creator** controls the extent to which AMACS promotes artifacts generated by agents who have generated other popular artifacts.
- **Gratitude** controls an agent A's willingness to view artifacts generated by agent B because B has done the same for A in the past. Note that this is highly related with the concept of Reciprocity introduced in the previous section, but in AMACS Gratitude only becomes Reciprocity when the reciprocated reviews end up being positive.
- **Recommender Ranking** controls an agent's willingness to evaluate the first artifacts that are recommended to them as opposed to considering many options.

We experiment with the effects of each of these hyperparameters and show how they can induce specific behaviors later in the paper.

AMACS Instantiations

In order to demonstrate the diversity of tasks to which AMACS can be applied we present three instantiations, each focused on a different creative discipline. Creating a new AMACS instantiation is as simple as implementing the agents' generation process and defining an artifact space by writing evaluation functions. The examples provided here are relatively simple, allowing us to focus on the social mechanics of AMACS rather than the generation and evaluation details which will vary from application to application. AMACS is equally capable of modeling interactions between creative agents with more sophisticated processes than those described here.

We provide only brief descriptions of the creative tasks and the evaluation functions used in each instantiation. Further details on how agents generate and evaluate artifacts can be found in the AMACS code repository.

AMACS for Image Generation In this AMACS instantiation agents generate 16x16 grayscale images. All agents try to generate images that are symmetrical and that have a small cross pattern in any of the four corners. We note that these constraints are arbitrary and were chosen only to demonstrate the idea of a universal aesthetic standard amongst agents. There are two dimensions along which agents can choose their tastes: the overall brightness of the image and the average contrast between columns in the image. Generation is accomplished using a genetic algorithm.

AMACS for Title Generation In this AMACS instantiation agents generate plausible titles for academic papers. "Plausibility" was measured using two methods: a neural network trained on 46,198 examples, and hand-coded rules designed to detect failure modes of the neural network. Agents chose their own tastes along three dimensions, each measuring the degree to which artifacts belong in one of three subclasses: Computer Science papers, Medicine papers, and Humanities papers. Each dimension is measured with an LSTM trained to detect whether a title belongs to the corresponding subclass. We train the LSTMs separately (though on overlapping data) so that a single title could theoretically score high on all three classifiers, although it is easier to earn a high score on just one. See Elgammal et. al. (2017) for a demonstration of why subclass membership is a powerful consideration for creative agents. Title generation is accomplished using a genetic algorithm.

AMACS for Policy Generation Agents in this AMACS instantiation create policy look-up tables for a robot control problem inspired by (Mitchell 2009, p. 130–142). In

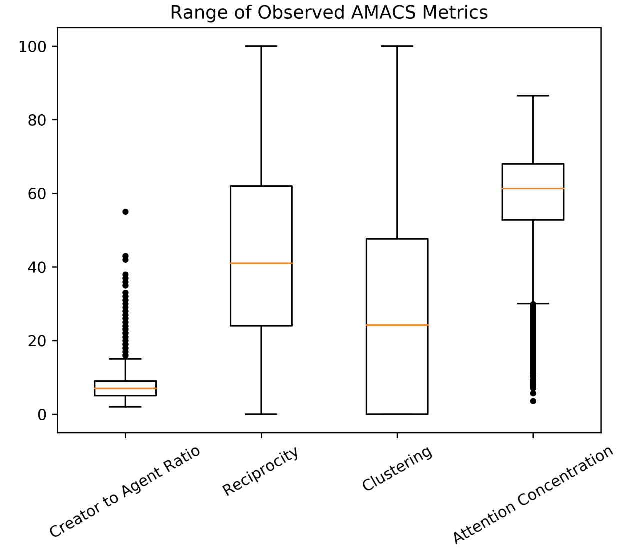


Figure 2: Full range of behavior observed in AMACS using all three instantiations and both SPM and TPM. AMACS demonstrates considerable flexibility in Reciprocity, Clustering, and Attention Concentration. It has limited flexibility in the Creator to Agent Ratio metric.

our version of this problem, a simulated robot lives in a 4 x 4 grid with blue and red trash scattered throughout. The AMACS agents compose instructions for the robot on how to navigate through the world and collect trash. All agents want to help the robot avoid running into walls, and each agent gets to choose the percentage of red and blue trash the robot collects. Agents generate policies using a genetic algorithm. We also implemented a Monte Carlo Tree Search approach for policy generation, but found that it was slower and caused agents to be less satisfied with their own artifacts.

Demonstrating the Flexibility and Applicability of AMACS

Given the human social behavior data and the simulated networks described earlier, we are ready to quantitatively validate that AMACS is capable of exhibiting human-like social behavior. Specifically, in this section we will demonstrate that by manipulating AMACS hyperparameters we can induce a wide range of behaviors, including the behaviors observed in human communities. Our purpose is not to demonstrate that AMACS always acts the same way as human or responds to stimuli in the same way as humans; rather, we seek to show that AMACS has enough flexibility that it can be coaxed into demonstrating the same network-level behavior as specific human social networks.

Experiment Setup

We discover the range of possible AMACS behavior with two sets of experiments in which we manipulate the 11 hyperarameters described in the previous section. In the first set of experiments, we change one hyperparameter's value at a time while keeping all other hyperparameter values fixed, which we refer to as Single Parameter Modulation (SPM). In the second set of experiments we introduce more noise by randomly and independently modulating the values of all 11 hyperparameters simultaneously, which we refer to as Total Parameter Modulation (TPM).

To perform SPM, we define a set of hyperparameter values V = {-20, -10, -1.0, -0.5, 0.0, 0.5, 1.5, 2.0, 10, 20, 30, 40}. The purpose in selecting these specific values is to measure what happens when we go far below, slightly below, slightly above, and far above a default value of 1.0. We refer to the set of all hyperparameters as P. For each hyperparameter $p \in P$ and each hyperparameter value $v \in V$, we produce a combination c of network inputs where p is set to v and all other hyperparameters are set to 1.0. For each generated combination c, we run all 3 AMACS instantiations 4 times, after which we record the four resultant network-level metrics. In total this requires 1,584 network runs.

To perform TPM, we split V into two subsets, V_S = {-1.0, 0.5, 0.0, 0.5, 1.0, 1.5, 2.0} and V_L = {-20, -10, 0, 10, 20, 30, 40}, where S and L stand for "small" and "large" and refer to the magnitudes of the included values. For each subset, we generate 700 random combinations of hyperparameter values in which each value is used 100 times for each hyperparameter. The purpose of splitting V into two subsets for TPM is to avoid situations in which large value changes in one hyperparameter drown out small value changes in other hyperparameters, i.e. we first modulate all hyperparameters on a small scale and then again on a large scale. We run each combination of hyperparameters for 30 generations each on all 3 AMACS instantiations. In total this involves 4,200 network runs.

Between SPM and TPM we perform a total of 5,784 network runs. Collectively these give us a broad understanding of the types of behavior AMACS is capable of modelling.

Simulation Results

Figure 2 shows the full range of metric values observed in all AMACS runs. We can see that AMACS is remarkably flexible with respect to observed Reciprocity and Clustering values; AMACS has produced the full range of possible values, and the spread is wide enough that no possible value can be classified as an outlier. AMACS also exhibits a fairly wide Attention Concentration spread, with values ranging from 3.6 to 86.5 including outliers. AMACS appears to be the least flexible in its Creator to Agent Ratio (CAR). The vast majority of AMACS runs had CARs less than 20, and even the highest magnitude outlier is only 55.5. This is lower than the r/ArtCrit CAR, meaning that some human behavior is outside the range of what AMACS can produce, at least with the instantiations and hyperparameters tested here. Future efforts to model a wider spread of CAR behaviors may consider changing the rules for how agents choose whether to be creators.

In order to validate AMACS's relevance as a tool for modelling human behavior, we compare AMACS runs to the human communities analyzed (Scratch, FanFiction, and r/ArtCrit). For each community we find the AMACS run which was the most similar to human behavior in each indi-

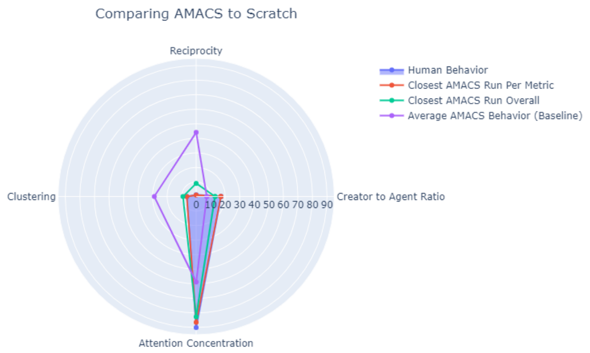


Figure 3: Comparison of AMACS behavior to the Scratch community. The blue shaded area represents human behavior. The red and green lines show the AMACS runs most similar to the human community in each individual metric and over all four metrics, respectively. The purple line shows average AMACS behavior and is included for reference.

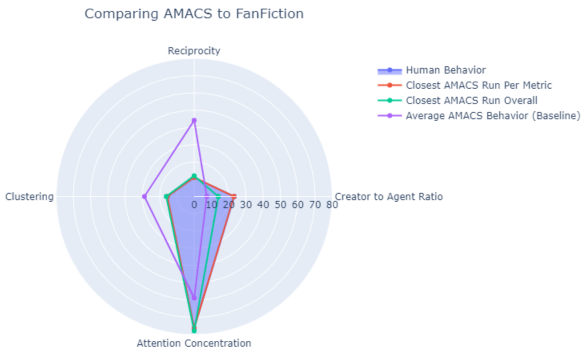


Figure 4: Comparison of AMACS behavior to the FanFiction community. The blue shaded area represents human behavior. The red and green lines show the AMACS runs most similar to the human community in each individual metric and over all four metrics, respectively. The purple line shows average AMACS behavior and is included for reference.

vidual metric and which was the most similar over all four metrics (measured with Euclidean distance). These results are visualized in Figures 3-5. We see that AMACS does fairly well at replicating the behavior of the Scratch and FanFiction communities, including nearly matching Scratch's remarkably high Attention Concentration. It is less successful at replicating r/ArtCrit's behavior, particularly in the CAR metric which, as noted earlier, is where AMACS is currently the least adaptable. AMACS is largely able to replicate the behavior of these three communities, indicating that it will likely be successful at modelling many other human creator networks.

Implications for Human Creators

The described parameter modulation experiments demonstrate the range of possible AMACS behaviors, but they also enable us to analyze the quantitative relationships between each hyperparameter and each network-level metric. Understanding these relationships is helpful for future AMACS designers hoping to induce specific behaviors from automated agents. This information can also help administrators of human creative communities to maximize the experiences of their members, provided that AMACS trends hold for human communities as well. Trends found in AMACS are not guaranteed to exist in human communities, but they indicate possibilities that may warrant further investigation.

To analyze the effects of each hyperparameter, we find the Pearson correlation between each hyperparameter and each network-level metric over all 5,784 network runs described above. Results are shown in Figure 6.

The strongest correlation observed is between the Popular Artifact hyperparameter (AMACS's tendency to promote popular content) and Attention Concentration. This is unsurprising, as recommending popular artifacts creates a positive feedback loop that keeps a few artifacts at the center of attention. This relationship matches the recommendation in (Xu and Bailey 2012) that administrators of online communities of creators can spread attention by increasing the personalization of user's 'Browse' or 'Explore' pages, as opposed to only recommending globally popular artifacts. New Artifact (AMACS's tendency to promote new content) shows a strong negative correlation with Attention Concentration, indicating another possible way that online platforms could spread attention when increased personalization is not possible.

The strongest indicator of a network's Reciprocity is the Gratitude hyperparameter (which represents how willing an agent is to review a peer's work because that peer has given positive reviews in the past). If administrators of online platforms want to increase the reciprocity of their communities, they might consider adding features that encourage gratitude, such as notifying users of generous actions and encouraging them to return the favor. For example, when a Reddit user receives a new follower, they receive a notification saying "[USERNAME] just followed you. Go check them out to learn more about them." This type of call to action encourages gratitude and, by extension, reciprocity.

For Clustering, the strongest indicator is the Mutual Contact hyperparameter (which controls an agent's desire to view artifacts created by agents with whom they share a mutual contact). There are two ways an administrator of an online social platform might use this information to increase Clustering. The first is by explicitly calling out the existence of mutual contacts in the site's UI. Facebook does this by listing the number of mutual friends user's have with each other, encouraging users with many mutual friends to connect. The second, more subtle approach is to use mutual contacts in determining which artifacts to recommend to a user on their "Browse" or "Explore" pages, which many social media sites already do.

Perhaps the most surprising strong correlation is between

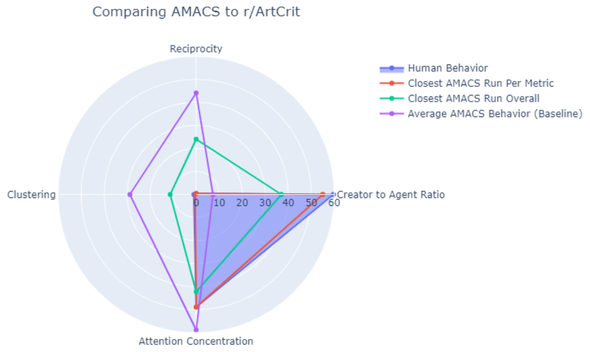


Figure 5: Comparison of AMACS behavior to the r/ArtCrit community. The blue shaded area represents human behavior. The red and green lines show the AMACS runs most similar to the human community in each individual metric and over all four metrics, respectively. The purple line shows average AMACS behavior and is included for reference.

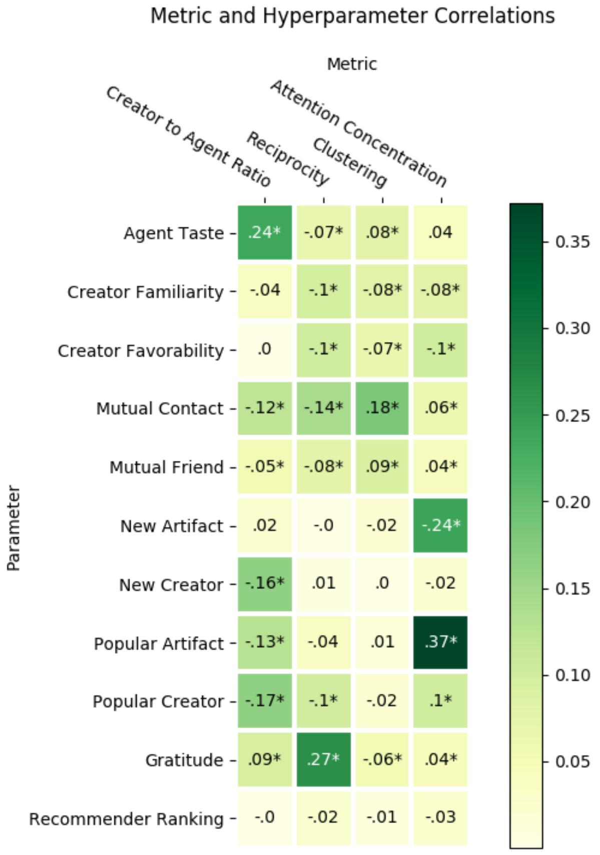


Figure 6: Pearson correlation between each hyperparameter and metric in AMACS. Darker colors indicate larger magnitudes, and $*$ indicates significant relationships ($\alpha = 0.01$).

Agent Taste (the personalization of AMACS's artifact recommendations) and Creator to Agent Ratio. AMACS agents become creators when they are confident in their own tastes, so the most likely reason for this correlation is that increased personalization leads to increased confidence, as agents consistently find artifacts that reinforce their current tastes.

Out of the eleven hyperparameters tested, ten showed statistically significant correlations with at least one metric, and 8 showed significant correlations with more than one metric.

We look forward to future work that may validate the degree to which these trends hold for human societies and discover other ways in which modelling tools can help inform our understanding of human behavior.

Ethical Considerations

One might reasonably ask if it is wise to study the ways in which community administrators can induce desired behaviors in their communities, as this might be interpreted as manipulation. The authors of this paper believe that studying the power of platform administrators in a public and academic setting adds transparency and accountability to the larger discussion of ethical platform administration. Design decisions affect users whether we understand their effects or not; this line of research empowers administrators to be more deliberate and thoughtful with the influence they already have. It is our hope that educating both users and administrators will help both parties make decisions that are beneficial to everybody.

Conclusion

In this work we have introduced a data-driven and task-agnostic architecture for modelling the social behavior of creative agents. We have studied real-world communities of creators and replicated many of their behaviors in an automated setting using AMACS: the Architecture for Multi-Agent Creative Societies. AMACS is designed to be flexible and user-friendly, and we hope it will provide a useful test bed and common setting for future experiments. Future areas of improvement could include defining more robust and descriptive metrics for understanding network-level social behavior, collecting data on more human creator communities, and investigating the experience of individual network participants rather than analyzing aggregated data.

We look forward to future work that will use socialization both to improve the efficacy of artificial creative agents and also "to contribute to the understanding of human creativity" (Saunders and Bown 2015). Learning from from human behavior, as we have done here, has the potential to improve our models and the performance of computational creativity systems. Using real-world data to validate automated systems also allows information to flow the other way; phenomena that emerge in our simulations give us clues about how human creativity may work. We hope that this and future work continues to improve the experience of human creators and the performance of automated ones.

Author Contributions

Berkeley Andrus wrote the paper and designed and conducted the experiments described, with Nancy Fulda advising and editing.

Acknowledgements

We would like to thank Dan Ventura for his advice and feedback that contributed to the design of AMACS. We would also like to thank Jacob Crandall for his advice throughout the project and for inspiring the four metrics used in our analysis.

References

[Alnajjar and Hämäläinen 2018] Alnajjar, K., and Hämäläinen, M. 2018. A master-apprentice approach to automatic creation of culturally satirical movie titles. In *Proceedings of the 11th International Conference on Natural Language Generation*, 274–283.

[Andrus 2021] Andrus, B. 2021. Modeling user relationships in online communities of creators. Master's thesis, Brigham Young University.

[Boden 1992] Boden, M. 1992. *The Creative Mind*. London: Abacus.

[Boyd, Hushlak, and Jacob 2004] Boyd, J. E.; Hushlak, G.; and Jacob, C. J. 2004. Swarmart: Interactive art from swarm intelligence. In *Proceedings of the 12th Annual ACM International Conference on Multimedia*, 628–635.

[Campbell et al. 2016] Campbell, J.; Aragon, C.; Davis, K.; Evans, S.; Evans, A.; and Randall, D. 2016. Thousands of positive reviews: Distributed mentoring in online fan communities. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, 691–704.

[Centola 2010] Centola, D. 2010. The spread of behavior in an online social network experiment. *science* 329(5996):1194–1197.

[Chang et al. 2020] Chang, J. P.; Chiam, C.; Fu, L.; Wang, A.; Zhang, J.; and Danescu-Niculescu-Mizil, C. 2020. Convokit: A toolkit for the analysis of conversations. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 57–60.

[Crain and Bailey 2017] Crain, P. A., and Bailey, B. P. 2017. Share once or share often? exploring how designers approach iteration in a large online community. In *Proceedings of the 2017 ACM SIGCHI Conference on Creativity and Cognition*, 80–92.

[Csikszentmihalyi 2014] Csikszentmihalyi, M. 2014. Society, culture, and person: A systems view of creativity. In *The Systems Model of Creativity*. Springer. 47–61.

[Dorfman 1979] Dorfman, R. 1979. A formula for the gini coefficient. *The review of economics and statistics* 146–149.

[Elgammal et al. 2017] Elgammal, A.; Liu, B.; Elhoseiny, M.; and Mazzone, M. 2017. Can: Creative adversarial networks generating "art" by learning about styles and deviating from style norms. In *8th International Conference on Computational Creativity, ICCC 2017*. Georgia Institute of Technology.

[Evans et al. 2017] Evans, S.; Davis, K.; Evans, A.; Campbell, J. A.; Randall, D. P.; Yin, K.; and Aragon, C. 2017. More than peer production: Fanfiction communities as sites of distributed mentoring. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, 259–272.

[Gini 1912] Gini, C. 1912. Variabilità e mutabilità. *Reprinted in Memorie di Metodologica Statistica (Ed. Pizetti E.*

[Glăveanu 2013] Glăveanu, V. P. 2013. Rewriting the language of creativity: The five a's framework. *Review of General Psychology* 17(1):69–81.

[Gómez de Silva Garza and Gero 2010] Gómez de Silva Garza, A., and Gero, J. S. 2010. Elementary social interactions and their effects on creativity: A computational simulation. In *ICCC*, 110–119. Citeseer.

[Granovetter 1973] Granovetter, M. S. 1973. The strength of weak ties. *American journal of sociology* 78(6):1360–1380.

[Greenfield and Machado 2009] Greenfield, G., and Machado, P. 2009. Simulating artist and critic dynamics. In *Proceedings of the International Joint Conference on Computational Intelligence, Funchal, Madeira, Portugal, October*, 5–7.

[Hämäläinen, Alnajjar, and others 2019] Hämäläinen, M.; Alnajjar, K.; et al. 2019. Modelling the socialization of creative agents in a master-apprentice setting: The case of movie title puns. In *Proceedings of the 10th International Conference on Computational Creativity*. Association for Computational Creativity.

[Hantula and Linkola 2018] Hantula, O., and Linkola, S. 2018. Towards goal-aware collaboration in artistic agent societies. In *Proceedings of the Ninth International Conference on Computational Creativity ICCC 2018, Salamanca, 25-29 June*. Association for Computational Creativity (ACC).

[Jackson 2019] Jackson, M. O. 2019. *The Human Network: How Your Social Position Determines Your Power, Beliefs, and Behaviors*. Vintage.

[Jennings 2010] Jennings, K. E. 2010. Developing creativity: Artificial barriers in artificial intelligence. *Minds and Machines* 20(4):489–501.

[Linkola and Hantula 2018] Linkola, S., and Hantula, O. 2018. On collaborator selection in creative agent societies: An evolutionary art case study. In *International Conference on Computational Intelligence in Music, Sound, Art and Design*, 206–222. Springer.

[Malik and Mucha 2013] Malik, N., and Mucha, P. J. 2013. Role of social environment and social clustering in spread of opinions in coevolving networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 23(4):043123.

[Marlow and Dabbish 2014] Marlow, J., and Dabbish, L. 2014. From rookie to all-star: Professional development in a graphic design social networking site. In *Proceedings of the*

17th ACM Conference on Computer Supported Cooperative Work & Social Computing, 922–933.

[Milli and Bamman 2016] Milli, S., and Bamman, D. 2016. Beyond canonical texts: A computational analysis of fanfiction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2048–2053.

[Mitchell 2009] Mitchell, M. 2009. *Complexity: A Guided Tour*. Oxford University Press.

[Pace et al. 2013] Pace, T.; O'Donnell, K.; DeWitt, N.; Bardzell, S.; and Bardzell, J. 2013. From organizational to community creativity: Paragon leadership & creativity stories at etsy. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, 1023–1034.

[Pérez y Pérez et al. 2010] Pérez y Pérez, R.; Negrete, S.; Peñalosa, E.; Ávila, R.; Castellanos-Cerda, V.; and Lemaitre, C. 2010. Mexica-impro: A computational model for narrative improvisation. In *ICCC 2010*, 90–99.

[Saunders and Bown 2015] Saunders, R., and Bown, O. 2015. Computational social creativity. *Artificial life* 21(3):366–378.

[Sylvan 2007] Sylvan, E. 2007. *The Sharing of Wonderful Ideas: Influence and Interaction in Online Communities of Creators*. Ph.D. Dissertation, Massachusetts Institute of Technology.

[Sylvan 2010] Sylvan, E. 2010. Predicting influence in an online community of creators. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1913–1916.

[Ventura 2019] Ventura, D. 2019. Autonomous intentionality in computationally creative systems. In *Computational Creativity*. Springer. 49–69.

[Wiggins 2006] Wiggins, G. A. 2006. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19(7):449–458. Creative Systems.

[Wright, Purver, and others 2020] Wright, G.; Purver, M.; et al. 2020. Creative language generation in a society of engagement and reflection. In *Proceedings of the Eleventh International Conference on Computational Creativity*. Association for Computational Creativity (ACC).

[Xu and Bailey 2012] Xu, A., and Bailey, B. 2012. What do you think? a case study of benefit, expectation, and interaction in a large online critique community. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, 295–304.

Noise as a Key Factor in Realizing a Creative Society

Guanhong Li and **Xiaoyun Guo** and **Takashi Hashimoto**
School of Knowledge Science
Japan Advanced Institute of Science and Technology
Nomi, Ishikawa 923-1292, Japan
{gu_li@kufs.ac.jp, xiaoyun@jaist.ac.jp, hash@jaist.ac.jp}

Abstract

To achieve a creative society, it is important to maintain a diversity of values. However, due to automatic alignment of values, maintaining diverse values is a challenge. We propose that the noise, i.e, the uncertainty in understanding others' values, may play a key role in realizing a creative society by moderating values alignment. In addition, the potential effects of noise may be affected by the social perception bias at the societal level. To study the dynamics of values diversity resulting from peer interactions with noise and perception bias, we present a hybrid opinion dynamics model, in which values is represented as both continuous and discrete categories. We simulate online social networking with various levels of noise and perception bias. We found that the positive effects of noise are twofold: it helps to moderate the alignment process of values within a group and to improve the social inclusiveness of extremism views. We conclude that noise contributes to preventing social fragmentation and monolithicization, and therefore plays a key role in realizing a creative society.

Introduction

A vision of the future of our society is the "Creative Society", in which the co-creation, instead of economy, is considered to be the core of the social systems (Iba 2016). Co-creation is a creative activity that requires collaboration between multiple individuals. Previous researchers noted that including members with diverse backgrounds in creative activities can create a more inclusive atmosphere and lead to better outcomes (Mannix and Neale 2005; Hawlina, Gillespie, and Zittoun 2019). At the societal level, maintaining a diversity of values (i.e., ideological beliefs such as conservative/liberal) could also create an inclusive social atmosphere and benefit social co-creation. Therefore, maintaining diverse values can facilitate the realization of a creative society.

Meanwhile, maintaining diverse values is a challenge due to social alignment. In social-cognitive science, social alignment refers to the alignment of minds and bodies in social interaction (Gallotti, Fairhurst, and Frith 2017), which can be an automatic and unconscious process (Chartrand and Bargh 1999). When understand others' values, unconscious alignment of values may occur, negatively affecting the diversity of values.

We propose that one effective factor that can counteract the negative effects of social alignment is the noise, i.e., the uncertainty, when understanding the values of others. In a modern society, we understand others' values mainly through their messages posted on online social networks sites (SNS). Usually, the noise is considered negative, as it can produce misunderstanding and cause severe problems in many cases. Hence, a goal of technology development (e.g., annotation, labeling) is to reduce the influence of noise (Souri, Hosseinpour, and Rahmani 2018). However, too low noise may strengthen the values alignment and thus have negative impacts on realizing a creative society.

This study aims to clarify the potential role of noise in maintaining diverse values. We hypothesize that the role of noise is affected by the societal level of social perception bias. Social perception bias refers to one's bias when perceiving the values of others, which can have different directions related to social categorization (e.g., distinguishing between *ingroup* and *outgroup* (Brewer 1999)). The societal level of such a bias could be considered as a distinguishing characteristics varied by culture and society. In a society with high-level social perception bias, the understanding of others is more easily biased by the perceiver's own values. Social perception bias can affect the understanding of others' values and therefore can play a role in the alignment of values.

This work focuses on the potential effects of online social networking on the alignment of values. We followed the KISS (Keep It Simple, Stupid) principle to study computational social creativity (Saunders and Bown 2015). We developed a multi-agent model for modeling online social networking and the alignment of values in societies with various bias levels. In our model, each agent has its values, represented in a hybrid way, i.e., as both a continuous numerical value and a discrete category (e.g., Leftism/Centrism/Rightism in politics, Centralization/Neutral/Localism in digital products/services). The agents are uniformly distributed in a virtual space and read messages posted by their neighbours who have similar service networking preferences (e.g., the frequency of using specific social networking sites). By reading a message, an agent perceives the values of the message poster and may adjust its own values accordingly. We modeled the values with well-established categories and remains stable at the individual level (e.g., individual attitudes towards certain policy issues (Carsey and Layman 2006)), which is consistent with the social intuitionist view (Haidt 2001). Hence, the social alignment is bounded, i.e., an agent only align its own values within the same category and thus always stick to that category.

We simulated online social networking under different levels of noise, which is related to the development of information technology. We analyzed the dynamics of values diversity and discussed potential role of noise.

Model

This work presents an opinion dynamics (OD) model, which are agent-based models adapted from physics to study the formation of opinions or beliefs. We used a hybrid design, i.e., the opinions are represented in both discrete and continuous ways, similar to the three-state CODA (Continuous Opinions and Discrete Actions) model (Martins 2010). To better reflect the nature of social values, we made two major changes. First, the agents in the CODA model perceive others' opinions as discrete numbers, while we use a continuous design. Second, the updates of the opinions is bounded by the agents' initial value categories. The fixed-category design enables the study on the interactions between groups with well-established opinion categories.

In our model, multiple agents are aligned in 1-dimension space, which is the simplest case of uniformly distributed social networks. The position of agents is fixed during the simulation. Each agent has an opinion about the sense of values. For an agent i, its values is represented as both a continuous numerical value v_i and a discrete category o_i. The v_i is represented as a real number within the interval $(0, 1)$. The o_i is represented as an element in the finite set $\{1, 2, 3\}$, corresponding to three positions of values: the two ends of the spectrum (i.e., $v_i \leq 1/3$ and $v_i > 2/3$) and the neutral position (i.e., $1/3 < v_i \leq 2/3$). The category o_i is fixed in a simulation.

At each time step, every agent reads a message posted by a random neighbor within a distance L, which represents the differences in their social networking preferences. We set $L = 20$ in this work. As we use a 1-dimension setting, this setting means that an agent can see the posts made by 20 other agents with closest social networking preferences.

When the agent i reads a message by the neighbor j, the values of j is perceived by i as both a numerical value v'_j and a category o'_j. The v'_j is computed first, and then the o'_j is set accordingly. The v'_j is a random value following the uniform distribution within the interval D_{ij}, which is determined by the true value (v_j), the noise level (B), and the perceiver's bias towards the poster j (δ_{ij}):

$$D_{ij} = (v_j - B + \delta_{ij}, v_j + B + \delta_{ij}) \tag{1}$$

The parameter B is a variable within the interval $(0, 0.5)$, which produces a random offset from the true value. The setting of $B > 0$ corresponds to inevitable noise in the understanding of others. When $B \geq 0.5$, the values corresponding to the neutral position 0.5 would be perceived as any values in the whole space $[0, 1]$.

The perceiver bias produces a biased offset from the true value, which is related to the distance between the values v_i and v_j:

$$\delta_{ij} = \begin{cases} |v_i - v_j|^{1/k} & \text{if } v_i - v_j < -\lambda, \\ & \text{or } 0 \leq v_i - v_j < \lambda \\ -|v_i - v_j|^{1/k} & \text{if } -\lambda \leq v_i - v_j < 0, \\ & \text{or } \lambda \leq v_i - v_j \end{cases} \tag{2}$$

, where the λ is a threshold for determining the bias direction, and k represents the societal level of social perception bias. The threshold λ represents the boundary between *ingroup* and *outgroup* categorization (see Figure 1, the dashed line). When the distance between v_i and v_j is smaller than λ (on the left of the dashed line), v'_j tends to be similar to v_i. Otherwise, v'_j tends to be different from v_i. In this study, we set $\lambda = 0.5$, which corresponds to a balanced categorization when the regions of inclusions and exclusions are symmetrical.

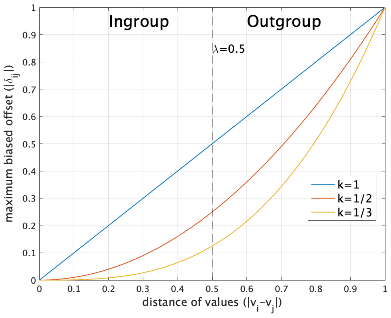


Figure 1: The effect of the bias level k. Dashed line shows the threshold λ.

The parameter k is a variable within the interval $(0, 1)$, which determines the relationship between the values differences and the biased offset (see Figure 1). A smaller k corresponds to a society where the understanding of others is less susceptible to the perceiver's own values.

Following the computation of v'_j and o'_j, if the message poster j appears to belonging to the same category as the perceiver i, i.e., $o_i = o'_j$, i would adjust its values to align with j. First, the numerical value at the time step t is computed as:

$$v_i(t + 1) = v_i(t) - \mu(v_i(t) - v'_j(t)) \tag{3}$$

, where μ defines the speed of alignment, which is set to 0.3 (a moderate speed) in this study. The updates of v_i follows a bounded design, i.e., i will do nothing if $o_i \neq o'_j$. Hence, i will never change the category o_i, and the numerical value v_i will be kept within the boundary of the initial category.

Simulation Results

Using the proposed model, we run simulations with various settings of noise level (B) and bias level (k). In all runs, the number of agents (N) was 3000, which were divided equally into three groups (Group A, B, and C). At the beginning of each run, for the agents in Group A, B, and C, the initial category (o_x) was fixed to 1, 2, and 3, respectively, while the initial numerical value (v_x) was set to random numbers following a uniform distribution in the interval $(0, 1/3)$, $(1/3, 2/3)$, and $(2/3, 1)$, respectively.

A qualitative analysis is performed to examine potential effects of noise on the society behavior at low- and high-level of bias. We found that low noise affects the behavior of low- ($k = 0.33$) and high-biased ($k = 0.90$) societies in different ways. The changes of population histogram in typical runs at various noise levels (0.01, 0.05, 0.10, and 0.30) are plotted in Figure 2 and Figure 3, corresponding to the low- and high-level bias settings, respectively.

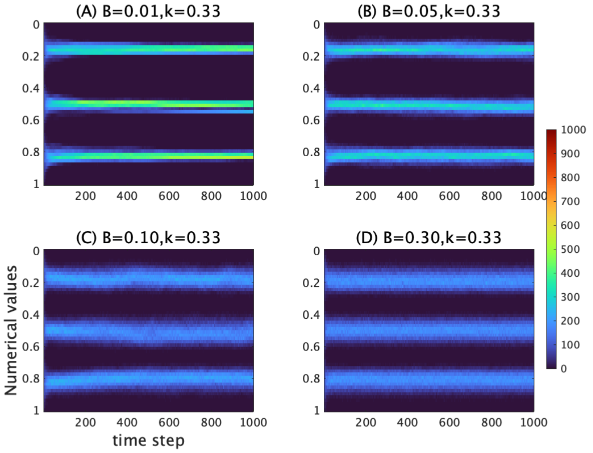


Figure 2: Changes of population histogram in typical low bias ($k = 0.33$) runs. Four panels corresponds to various noise levels (B). Color bar indicates the population.

At the low bias level, a decrease in noise level leads to a convergence of each group's values toward the center of respective groups. At the high bias level, the results are different among groups. For the extremism groups (Group A and C), a decrease in noise level did not only lead to a convergence of values towards the group center, but also bring the distance between extremism groups closer and thus result in a narrower range of values. For the centrism group (Group B), however, diverse values could be observed at low noise levels.

In the quantitative analysis, we first analyzed the overall diversity of values by computing the Shannon Index (e) of the numerical value (v_x) for the whole population in the society. To compute e, we divided the interval $(0, 1)$ into 50 equal-width segments. Then the Shannon Index at the time step t was computed as:

$$e(t) = -\sum_{i=1}^{S} p_i(t) \log(p_i(t)) \qquad (4)$$

, where S is the segments space covering $(0, 1)$, and p_i is the number of agents who held values with a numerical value within each segment.

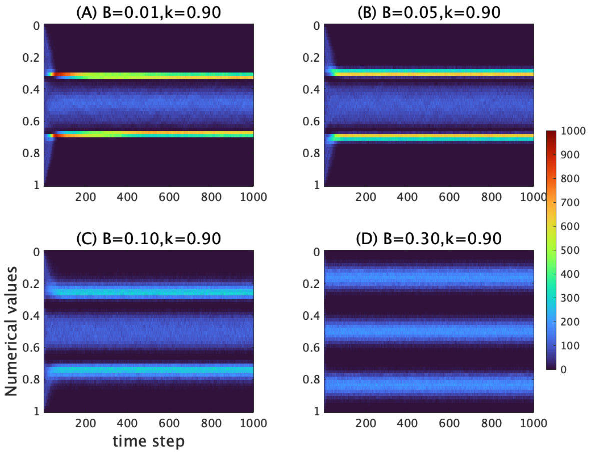


Figure 3: Changes of population histogram in typical high bias ($k = 0.90$) runs. Four panels corresponds to various noise levels (B). Color bar indicates the population.

Each run lasted for 1000 time steps. As showed in the results from typical runs, this setting is sufficient for the society to reach a stable state. We measured the final diversity in a run by the Shannon Index at the final step. We analyzed the averaged results of 30 runs.

For a low-level bias and a high-level bias settings, the overall diversity in relation to the noise level is plotted in Figure 4 *Left*. Within each group, the relationship between noise level and local diversity of values is plotted in Figure 4 *Middle*. At various noise levels, the effects of noise on the range of values (measured by the standard deviation) is plotted in Figure 4 *Right*.

Despite minor differences, the negative effects of low noise on the overall diversity did not differ noticeably in both magnitude and trend. In contrast, there is a difference in the noise effects on local diversity of values under different settings. At a low-level bias setting, low noise negatively affected local diversity in all three groups equally. At a high-level bias setting, the negative effects of low noise on local diversity in the extremism groups (Group A and C) was more drastic. For the centrism group (Group B), however, lowering noise could lead to a dramatic increase in the local diversity of values. Meanwhile, we also found a difference in the noise effects on the values range under different settings. At a low-level bias setting, low noise had little effects on the standard deviation of values. That is, at various noise levels, the range of values was maintained. At a high-level bias setting, lowering noise resulted in a decrease in the standard deviation of values. That is, the range of values became narrower at low levels of noise.

Discussion

In general, our results suggest that, regardless of the societal level of social perception bias, a certain degree of noise (i.e., uncertainty) when understanding others' values through online social networking is necessary for maintaining the di-

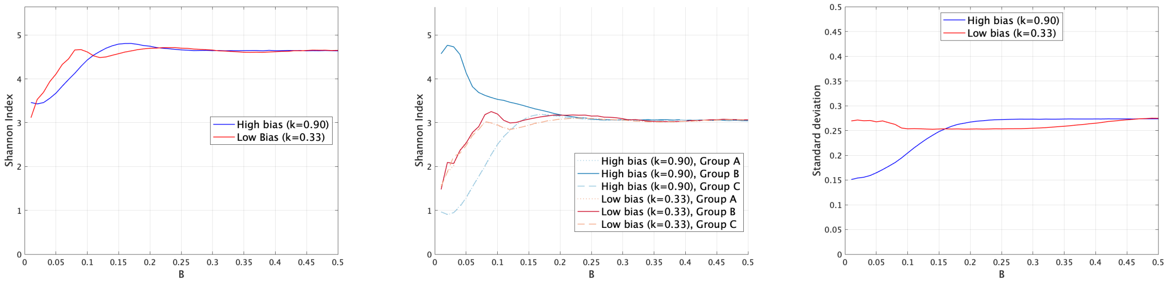


Figure 4: *Left*: Overall values diversity at different noise levels (B) under low- and high-level bias (k) settings. *Middle*: Local values diversity in each group in relation to noise (B) at low- and high-level bias (k). *Right*: Standard deviation of values in relation to noise (B) at low- and high-level bias (k).

versity of values, which is a key to achieving a creative society. Moreover, depending on the bias level of social perception in a society, the mechanism underlying the observed effect of noise may vary.

In societies with low levels of bias, there are fewer categorization errors caused by perceiver bias (see Figure 1). Values alignment occurs mainly within a group, while interactions between different groups are suppressed. In this case, low levels of noise can strengthen social fragmentation (see Figure 2). Thus, when the bias level of social perceiving is low, noise may be necessary for maintaining the overall values diversity by influencing the values alignment within a group, i.e., by maintaining local diversity of values.

For societies with high bias levels, since the perceiver's own values have a greater impact on the understanding of values, neighboring groups can influence each other. The alignment of values occurs not only within the groups, but also between the centrism groups and the groups on the two ends of the spectrum. This result is consistent with previous findings on the attraction from moderate opinions in the three-option CODA model (Martins 2010). We found that the inter-group attraction is strengthened at high bias level. Thus, a low level of noise can strengthen the social monolith (see Figure 3).

In a low-noise condition, the alignment of values reduces the internal diversity of the extremism groups. However, probably, for the centrism group, the effect of alignment can be counteracted by the attraction between neighboring groups, which could even lead to an increase in local diversity. Meanwhile, the attraction strengthened by low-noise leads to the convergence of the extremism groups toward the centrism group, thus narrowing the range of values. Therefore, when the bias level is high, noise not only affects the alignment process of values within groups, but also affects the interactions between neighboring groups, thus contributes to the overall diversity of values.

Our definition of noise and bias is compatible with the concepts proposed recently by Kahneman, Sibony, and Sunstein (2021). In their work, the noise in social perception was considered a "flaw" that should be reduced as much as possible. In contrast, this study shows that noise can effectively reduce the negative effects of social alignment on diversity of values, thus contributing to a more inclusive social atmosphere and facilitating the realization of a creative society. In particular, our results suggest that noise is more important for societies with high bias levels of social perception. In a high biased society, values alignment is more likely to occur, thus creating a social mainstream, making it much more difficult to maintain the range of values, i.e., the inclusiveness of society, without the help of noise.

Conclusion

Noise exists in the understanding of others' values. However, in modern online social networking, the noise has been greatly reduced by technological developments in the pursuit of communication accuracy. Through computer simulations, we demonstrated the downside of too little noise to realize a creative society, i.e., a reinforcement of social fragmentation and monolithicization. Our results suggest that noise plays a key role in maintaining the diversity of social values by preserving local diversity and social inclusiveness. In this sense, noise could be considered as a key factor in realizing a creative society.

It should be noted that this study only considered the effect of typical online social networking on the changes of values. It is unknown to what extent our results can be generalized to other forms of social interaction. Meanwhile, the formation and evolution of values would also be affected by other factors than social interaction. Therefore, the evidence and boundaries of the present findings need to be examined by future studies.

Nevertheless, this work adds to the literature emphasizing the importance of alternative information in social cognition (Salvi et al. 2021). In addition, this study contributes to future social creativity studies using simple and reproducible models of opinion dynamics. We demonstrated that opinion dynamics models can be used to study social creativity resulting from peer interactions. Future works interesting in the interactions between internal and external opinions could develop variants of our model using dynamic categories. We believe that these works could lead to a better understanding of the mechanisms of social creativity.

Author Contributions

GL: Conceptualization, Methodology, Software, Investigation, Visualization, Writing - original draft, review, and editing. XG: Conceptualization, Methodology, Investigation, Writing - review and editing. TH: Conceptualization, Methodology, Resources, Funding acquisition.

Acknowledgements

This work was supported by JSPS/MEXT KAKENHI Grant Number JP17H06383 (Innovative Areas #4903) and Kyoto University of Foreign Studies.

References

Brewer, M. B. 1999. The psychology of prejudice: Ingroup love or outgroup hate? *Journal of Social Issues* 55(3):429–444.

Carsey, T. M., and Layman, G. C. 2006. Changing sides or changing minds? party identification and policy preferences in the american electorate. *American Journal of Political Science* 50(2):464–477.

Chartrand, T. L., and Bargh, J. A. 1999. The chameleon effect: The perception–behavior link and social interaction. *Journal of Personality and Social Psychology* 76(6):893–910.

Gallotti, M.; Fairhurst, M.; and Frith, C. 2017. Alignment in social interactions. *Consciousness and Cognition* 48:253–261.

Haidt, J. 2001. The emotional dog and its rational tail: A social intuitionist approach to moral judgment. *Psychological Review* 108(4):814–834.

Hawlina, H.; Gillespie, A.; and Zittoun, T. 2019. Difficult differences: A socio-cultural analysis of how diversity can enable and inhibit creativity. *The Journal of Creative Behavior* 53(2):133–144.

Iba, T. 2016. Sociological perspective of the creative society. In Zylka, M.; Fuehres, H.; Fronzetti Colladon, A.; and Gloor, P., eds., *Designing Networks for Innovation and Improvisation*, Springer Proceedings in Complexity. Cham: Springer. 29–42.

Kahneman, D.; Sibony, O.; and Sunstein, C. R. 2021. *Noise: A Flaw in Human Judgment*. Little, Brown and Company.

Mannix, E., and Neale, M. A. 2005. What differences make a difference?: The promise and reality of diverse teams in organizations. *Psychological Science in the Public Interest* 6(2):31–55.

Martins, A. C. R. 2010. A middle option for choices in the continuous opinions and discrete actions model. *Advances and Applications in Statistical Sciences* 2(2):333–346.

Salvi, C.; Iannello, P.; Cancer, A.; McClay, M.; Rago, S.; Dunsmoor, J. E.; and Antonietti, A. 2021. Going viral: How fear, socio-cognitive polarization and problem-solving influence fake news detection and proliferation during covid-19 pandemic. *Frontiers in Communication* 5:562588.

Saunders, R., and Bown, O. 2015. Computational social creativity. *Artificial Life* 21(3):366–378.

Souri, A.; Hosseinpour, S.; and Rahmani, A. M. 2018. Personality classification based on profiles of social networks'users and the five-factor model of personality. *Human-centric Computing and Information Sciences* 8(1):24.

Bias and Creativity

Róisín Loughran
Regulated Software Research Centre
Dundalk Institute of Technology
Dundalk
Ireland
Roisin.loughran@dkit.ie

Abstract

This paper proposes a discussion on bias and its place in Computational Creativity research. Recent developments in Artificial Intelligence research have become more cognizant of the dangers and pitfalls in not recognising and addressing unseen biases within algorithmic systems. As many such methods are used for creative tasks, we propose that, as a community, we must consider bias possibilities and the implications they could have on the outputs and outcomes of research from this community.

Introduction

Despite many writings, experiments and discussions on the topic, Creativity is still a poorly defined concept. Trying to compute a poorly defined concept is immediately fraught with difficulties. Despite this persistent ambiguity, the field of Computational Creativity (CC) has been examining this problem for many years. Some of the main arsenal for this task have been methods and tools based in Machine Learning (ML) and Artificial Intelligence (AI). Such tools have been shown in recent years to be susceptible to various ethical issues, including, but not limited to detrimental bias. So we ask: Is bias within CC inevitable? And if it is, is this a bad thing?

This may be a complex question, but it is one worth considering. Despite academic and policy approaches to address Bias in AI as detailed in the following section, Big Tech have not always taken such matters as seriously as they should. While Google set up an Ethical AI Team in 2018, the controversial firing of Timnit Gebru in December 2019, and subsequent firing of Margaret Mitchell, who both co-led this Ethical AI Team, for refusing to withdraw a paper that criticised the use of large language models, demonstrates that this is a controversial topic that will not be easily solved (The Irish Times, 2021). Despite public outcry and an open letter of support for Gebru (Medium, 2020), Google did not reverse their decision, nor did they offer support in response to the harassment that subsequentially erupted towards the researchers on social media (Schiffer, 2021).

Bias, fairness and ethics are vitally important considerations for all applications of AI, and CC research is not exempt. In this short discussion paper, we propose a number of areas from which to consider bias in CC. First we consider bias, in its multiple forms and how it has been treated in AI. Then we look at the various types of algorithms that have been typically used in CC and consider if they are all as susceptible as each other. Finally we consider Creativity itself and how it may be rooted in biased decision making.

What is Bias?

As humans we all have inherent biases; when presented with a choice we have tendencies to lean towards one outcome, whether that is based on preference, exposure, belief or something intangible. Biases can be either conscious or subconscious. But, while the notion of *bias* invokes a very negative context, our innate biases are not inherently bad.

When it comes to trying to define bias, there appears to be no one standard clear definition. Dictionary definitions can include reference to prejudices: ‘Inclination or prejudice for or against one person or group, especially in a way considered to be unfair.’ (Lexico, 2022) or distortion: ‘Systematic distortion of results or findings from the true state of affairs, or any of several varieties of processes leading to systematic distortion.’ (*A Dictionary of Public Health*, 2007). Yet one of the seminal papers on bias in judgment refers to it as ‘..decisions based on beliefs concerning the likelihood of uncertain events’ (Tversky and Kahneman, 1974). Thus, a bias is simply a decision, one that is informed, either correctly or incorrectly, by some *a priori* belief or understanding we already possess. While the dictionary definitions define bias in terms of unfairness and distortion, the truth is that every day we use heuristics to make sense of the world around us. If we had no biases, our opinions of the world would be akin to white noise.

Detrimental Bias

Biases help us make decisions and form part of our personalities; it is when we encounter discriminatory bias that such judgments can be unfair, illegal or dangerous towards some in our society. As humans, we have inherent biases, and there is a strong potential for us to bring these biases into any algorithmic system we may create or deploy. The potential for algorithms to mirror human biases in decision making has been identified as one of the most straight forward ethical challenges in implementing AI in healthcare (Char, Shah and Magnus, 2018). For this reason there has been much academic research in to the types of biases that may be found or introduced to algorithmic systems in recent years (Mehrabi *et al.*, 2021) along with methods aimed to mitigate these effects (Bellamy *et al.*, 2019).

The problem of detrimental bias within AI systems is also increasingly being identified by regulatory authorities. NIST have recently published a standard on identifying and managing bias in AI (National Institute of Standards and Technology- US Department of Commerce, 2022) and IEEE plan to release the P7003 standard on Algorithmic Bias Considerations later this year (Koene, Dowthwaite and Seth, 2018). Bias, fairness and trustworthiness all contribute to the ethical implementation of AI. Ethics is an even larger consideration than that of bias, and many guidelines have been proposed to ensure ethical implementation of AI such as those proposed by the European Commission on the 'Ethics Guidelines for Trustworthy AI' (European Commission, 2019), although those proposed by the European Commission on are critical of these guidelines (Gille, Jobin and Ienca, 2020).

Fairness

If we remove all discriminatory biases from an algorithmic system we should be able to consider it fair. But, similar to bias, fairness is concept that is colloquially understood but difficult to universally define (Gajane and Pechenizkiy, 2017). Nevertheless many strides have been made to address fairness in AI including the development of fairness, accountability and transparency machine learning (FATML) (Veale and Binns, 2017). This study proposed three methods for addressing this: trusted third parties could be selective with data, online collaborative platforms with diverse organisations could promote fairness and unsupervised learning techniques could allow a fairness hypothesis be built for selective testing. Chen et al. noted that many ML models focus on balancing fairness and accuracy, but they argued that fairness should be evaluated in context of the given data and through data collection and study, rather than through constraint of the model (Chen, Johansson and Sontag, 2018). Binns further considered the nature of fairness and what it means for a ML algorithm to be fair by considering existing works on moral and political philosophy (Binns, 2018). This study questioned should fairness equate to equal opportunity for everyone or focus on minimising harm to the most marginalised. Such studies note that while many approaches to fairness in ML focus on data preparation, model-learning and use of the system, there is still much to be learned about the nature of fairness and discrimination before we can understand how applied ML can address this.

Algorithmic Bias

A variety of ML and AI techniques have been used to emulate Creativity over the years. Is any one more or less prone to bias than the other?

The Data-driven

The explosion of deep learning and in particular Convolutional Neural Nets (CNN) has been largely fueled by the creation of and accessibility to large image datasets. Such methods are commercially very favourable, but due to unbalanced, badly labelled datasets these are some of the most problematic systems in relation to detrimental bias. Birhane et al. discuss several dangers from ill-considered data curation practices including justice, consent and ethical transgressions (Birhane and Prabhu, 2021). Many detrimental biases are found to be discriminatory in relation to sensitive or protected characteristics such as race, gender etc. For this reason these characteristics are often not made available, although simply removing such characteristics from datasets has been shown to exacerbate rather than solve the issue, as latent relationships between other, non-sensitive attributes, can cause proxies that lead to the same biases (Chen *et al.*, 2019). Some methods have been proposed to use these proxies as a way to identify and mitigate against biasing against these characteristics (Lahoti *et al.*, 2020).

When considering bias in an AI system, the data does seem like the primary culprit as the source of bias; a system can only learn and reproduce the data and patterns it is given. But there are more aspects to consider. A recent systematic review found that Data-driven innovation (DDI) suffers from three major sources of bias: data bias, method bias and societal bias (Akter *et al.*, 2021). Thus, even in systems that are driven by the data, we should consider other internal design mechanisms and external influencing factors that can lead to detrimental bias.

The Evolutionists

Evolutionary Computation (EC) comprises a family of heuristic search methods based on Darwin's theory of survival of the fittest. A population of random solutions to a given problem is created and then iteratively improved ('evolved') over a series of generations. This improvement is driven by a fitness function – an evaluation measure of each individual derived by the creator of the algorithm. Such EC methods have been widely used in creative systems such as music, art and design (EvoSTAR, 2022).

EC systems may also work with large datasets, but there are further design decisions within their architectures that could lead to bias. Most notably it is the choice of fitness function that will dictate which individuals are deemed more fit and are hence given a better change of surviving to the next generation. This creates a statistical bias in favour of individuals that conform to the fitness defined. For objective, measurable tasks, this may be what is expected, but for subjective creative tasks, might this be creating an unwanted, or unexpected bias within the system?

Objective Search

Many other systems such as Generative Adversarial Networks (Elgammal *et al.*, 2017) among others have been used in the generation and study of creative artefacts and procedures. While they may differ in their architecture and style, one commonality among AI systems is that they each aim towards a specific objective. That may be to reduce an error, reach a goal or solve a problem, but a system must have an objective to train and aim towards.

The problem with such methods for creative tasks is that the best objective is not always easy to define. How would one pre-define the best melody, sketch or poem? A better search method may be to search for novelty rather than a pre-specified objective. Novelty search proposes that the optimal solution to a problem can be found when looking for a different solution or when looking for no particular solution at all (Lehman and Stanley, 2010). If you are searching for novelty, rather than an objective, it may be less likely that your search will be biased.

Creative Bias

The above may lead us to believe it is the AI, ML and computational tools we use that cause bias within a system. But what of the aesthetic, ever elusive, *Creativity* that we chase? Is Creativity itself susceptible to, or even dependent on, bias?

Like bias and fairness, Creativity is a concept that is understandable by most, yet hard to define in a generalized context. So, in effect, we are trying to ascertain if an ill-defined concept is susceptible to an undefined phenomenon. But, as noted above, we do have an innate understanding of what bias is and how it affects our judgments. In a similar manner we do have ways of measuring Creativity. It has been proposed that to identify Creativity the system must be able to display novelty and value (Boden, 1998).

Novelty At its most absolute meaning, novelty is an unbiased concept; either something is new or it is not. However, often what is meant by novelty is that it is new to the creator. An individual does not have to create something new to the world to have displayed creativity. Personal or P-Creativity is as valid as Historical H-Creativity. In this sense, the novelty of P-Creativity could be biased to the individual.

Value The value of a creative artefact is surely a biased measurement. The monetary value of an aesthetic artefact is measured by what the highest bidder is willing to pay for it. Such a measure is surely influenced by styles, fashion, popularity and a wealth of other immeasurable external biases, along with the internal biases of the buyers. Of course, the monetary value is only one, very superficial, measure of an artefact's worth. A generated piece may have artistic, academic, historical, personal or many other forms of value. But it is likewise difficult to imagine how such a measured value could be determined without any biases.

CC Evaluation

It has been noted numerous times, that evaluation does not take enough precedence in CC experiments (Jordanous, 2012). This is likely due to the complexity of defining what creativity is; how can one measure what you cannot define? Nevertheless, evaluation methods for creativity in computational systems have been proposed. However, many such methods center on human evaluations and judgments which are costly and may lead to limitations (Loughran and O'Neill, 2016). Using human evaluators is costly in both time and money. Furthermore, if we acknowledge that our human biases are subjective to our preferences then we must accept that any human evaluator will evaluate towards their own personal preferences. If someone is adjudicating the creativity of a music generation system, it is difficult to confirm they are judging the system on its creativity and not merely how much they like the melody it produces. When judging creative artefacts, humans tend to mistake what they subjectively 'like' for what it objectively 'good'.

For accurate human based evaluation, you must ascertain their expert knowledge in the given domain. For those judging music, for instance, you should determine how many years of formal music training they have had. Such data may help group certain subjects together, but you must acknowledge this training may not remove a bias but simply introduce new ones. Classically trained musicians may expect, and then favour, outputs of a high musical quality, or music technology students may expect high production value. Even the most experienced adjudicator is still subject to their own learned opinions and biases.

Crowdsourcing

With online resources, it is now quite simple and cost-effective to evaluate on a large cohort of people as Crowdsourcing platforms are increasingly being used for creative tasks (Oppenlaender *et al.*, 2020). However, using large, unregulated crowds to evaluate a creative artefact will surely introduce bias. If you are not sure what demographic your audience is from or what bias profile they have, how can you use their personal preferences as any evaluation of merit? If, instead of paying for a platform, you merely share an online evaluation survey yourself, you are introducing this into a personal circle of people who are, most likely, highly interested or trained in the specific field that you are interested in. In other words, if you created an online survey to evaluate your generated music, how would a random set of people around the globe judge this music in comparison to those on your Twitter feed?

Discussion

As noted earlier, bias is not an intrinsically bad word, or concept; our biases are simply based on heuristics that we need to make decisions. If we consider how we approach the development of a CC system, we must make a number of decisions before we even start development such as:

- The domain(s) within which we will develop and/or test the system;
- The representation used;
- The algorithm(s) employed;
- The validation method(s).

Each of these decision will be influenced by the developers education, experience, personal background and preferences. And many of these choices will require further, more intricate choices along the way – what genre of music will your system compose? What architecture will you use for implementation? Many of these choices are subjective and have no definitive best answer; we do not know the exact number of neurons an ANN must have to make a picture 'creative'. The fact is that we require the freedom to make these choices in order to have the scope to even investigate what it means to be creative. Our learned tendencies, preferences or *biases* may be necessary for us to find creativity in all the mundaneness out there.

In saying that, we know that AI will mimic human behaviors, even the worst of them. Therefore, it is still vitally important that we consider any harmful biases or discriminations that may be emulated by our systems. It is such detrimental biases that we must identify, evaluate and mitigate against.

Detrimental Bias in Creative Systems

We have considered biases in relation to CC in this paper, but where might the most detrimental biases be found in our community?

Demographic As a computer science field, we must acknowledge the lack of women represented in the CC community. Likewise, we must be aware of underrepresentation of other ethnical and minority groups. Such a homogenous demographic is missing out on significant potential contributions to our field. This is not an easy problem to tackle, nor is it unique to CC. However, active and meaningful steps aimed at increasing the diversity within CC research could only benefit the quality and range of our outputs. We would encourage the CC community to actively discuss what measures could be taken to address this.

Training Data Historically, artists have been predominantly male. Hence the training databases, in art, music etc., will have already been curated from a male-generated perspective. If a system is learning from data that has been created predominantly by men, then the female perspective within the training data is missing. It would be difficult to ascertain to what extent this may bias a system, but it is worth consideration. For example, in visual art, there is a strong bias towards the female nude form as opposed to the male form. While acceptable, typical or even encouraged in its day, this is certainly a bias in subject matter. In a similar manner, many training artefacts would be assumed to be biased towards Western style – unless the given study explicitly states otherwise.

Domain CC research can be undertaken in almost any problem domain, as many problems require critical, creative thinking. Despite the fact that much early research in creativity was illustrated using logical tasks, it has been noted that there has been a lack of studies on scientific and logical problems in more recent years (Loughran and O'Neill, 2017). If creativity is not dependent on the application domain, we must acknowledge that an over-representation in one domain over another may introduce a bias within the field in general. The consideration of new application fields may attract new researchers into the field and develop creativity research into new areas.

Complexity Systems that have more complex representation or require and utilise a lot of domain-specific information may appear more impressive and hence be judged to be more creative. We must ensure not to be biased towards more complex systems or become overly impressed by flashy displays.

Bias Types Mehrabi et al. identify 22 types of bias that can be found in ML systems (Mehrabi *et al.*, 2021). While there are many other works discussing types of bias that may be possible within such systems and, arguably, no such list could ever be exhaustive, this is an excellent resource to consider the types of biases your system may be susceptible to. When developing your creative system, it is worth reviewing each bias type to determine if your proposed system may be detrimentally susceptible to these, or other, biases.

Conclusions

As a field within AI, CC researchers should be aware of the possibilities and dangers that bias could pose to their work. This short paper is only intended to start the discussion around biases within CC systems and how we must be vigilant to recognise, acknowledge and, if necessary, mitigate against such biases. We recognise that, as humans, our biases form part of our personalities – our likes and dislikes lead us to make creative choices. We must assume that these biases can, and in some cases should, be passed on to the systems that we develop. These systems generate creative artefacts through the targets, fitness, datasets or benchmarks that we use in their development. We must be aware that the preferences and biases we have learned or inherently own, can be integrated, either consciously, or unconsciously, into our developed systems.

As scientists, we all wish for the most comprehensive, fair and accurate conclusions to our own undertakings. We can only achieve this if we ensure we question the decisions and assumptions we make, at each step of our own processes.

Author Contributions

R.L. ideated and wrote this paper alone.

Acknowledgements

This work has been carried out with the support of Dundalk Institute of Technology.

References

A Dictionary of Public Health (2007) *A Dictionary of Public Health*. Oxford University Press. doi: 10.1093/acref/9780195160901.001.0001.

Akter, S. *et al.* (2021) 'Algorithmic bias in data-driven innovation in the age of AI', *International Journal of Information Management*, 60, p. 102387. doi: 10.1016/J.IJINFOMGT.2021.102387.

Bellamy, R. K. E. *et al.* (2019) 'AI Fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias', *IBM Journal of Research and Development*, 63(4–5). doi: 10.1147/JRD.2019.2942287.

Binns, R. (2018) 'Fairness in Machine Learning: Lessons from Political Philosophy', *Proceedings of Machine Learning Research*, 81, pp. 1–11.

Birhane, A. and Prabhu, V. U. (2021) 'Large image datasets: A pyrrhic win for computer vision?', *Proceedings - 2021 IEEE Winter Conference on Applications of Computer Vision, WACV 2021*, pp. 1536–1546. doi: 10.1109/WACV48630.2021.00158.

Boden, M. A. (1998) 'Artificial Intelligence Creativity and artificial intelligence', *Artificial Intelligence*, 103.

Char, D. S., Shah, N. H. and Magnus, D. (2018) 'Implementing Machine Learning in Health Care — Addressing Ethical Challenges', *New England Journal of Medicine*, 378(11), pp. 981–983. doi: 10.1056/nejmp1714229.

Chen, I. Y., Johansson, F. D. and Sontag, D. (2018) 'Why is my classifier discriminatory?', in *Advances in Neural Information Processing Systems*, pp. 3539–3550.

Chen, J. *et al.* (2019) 'Fairness Under Unawareness: Assessing Disparity When Protected Class Is Unobserved ACM Reference Format'. doi: 10.1145/3287560.3287594.

Elgammal, A. *et al.* (2017) 'CAN: Creative adversarial networks generating "Art" by learning about styles and deviating from style norms', in *Proceedings of the 8th International Conference on Computational Creativity, ICCC 2017*.

European Commission (2019) *Ethics guidelines for trustworthy AI - Publications Office of the EU*. Available at: https://op.europa.eu/en/publication-detail/-/publication/d3988569-0434-11ea-8c1f-01aa75ed71a1 (Accessed: 11 April 2022).

EvoSTAR (2022) *EvoMUSART – EvoStar 2022*. Available at: http://www.evostar.org/2022/evomusart/ (Accessed: 12 April 2022).

Gajane, P. and Pechenizkiy, M. (2017) 'On Formalizing Fairness in Prediction with Machine Learning'. doi: 10.1145/3306618.

Gille, F., Jobin, A. and Ienca, M. (2020) 'What we talk about when we talk about trust: Theory of trust for AI in healthcare', *Intelligence-Based Medicine*, 1–2(June), p. 100001. doi: 10.1016/j.ibmed.2020.100001.

Jordanous, A. (2012) 'A Standardised Procedure for Evaluating Creative Systems: Computational Creativity Evaluation Based on What it is to be Creative', *Cognitive Computation 2012 4:3*, 4(3), pp. 246–279. doi: 10.1007/S12559-012-9156-1.

Koene, A., Dowthwaite, L. and Seth, S. (2018) 'IEEE P7003 standard for algorithmic bias considerations', *Proceedings - International Conference on Software Engineering*, (May), pp. 38–41. doi: 10.1145/3194770.3194773.

Lahoti, P. *et al.* (2020) 'Fairness without demographics through adversarially reweighted learning', *Advances in Neural Information Processing Systems*, 2020-Decem.

Lehman, J. and Stanley, K. O. (2010) 'Efficiently evolving programs through the search for novelty', in *Proceedings of the 12th Annual Genetic and Evolutionary Computation Conference, GECCO '10*, pp. 837–844. doi: 10.1145/1830483.1830638.

Lexico (2022) *BIAS | Meaning & Definition for UK English | Lexico.com*. Available at: https://www.lexico.com/definition/bias (Accessed: 31 March 2022).

Loughran, R. and O'Neill, M. (2016) 'Generative Music Evaluation: Why do We Limit to 'Human' ?', in *Conference on Computer Simulation of Musical Creativity*. Huddersfield. Available at: https://www.researchgate.net/publication/304284746 (Accessed: 12 April 2022).

Loughran, R. and O'Neill, M. (2017) 'Application domains considered in computational creativity', in *Proceedings of the 8th International Conference on Computational Creativity, ICCC 2017*.

Medium (2020) 'Standing with Dr. Timnit Gebru - Google Walkout for Real Change', *Medium*, 4 December, pp. 1–4. Available at: https://googlewalkout.medium.com/standing-with-dr-timnit-gebru-isupporttimnit-believeblackwomen-6dadc300d382 (Accessed: 11 April 2022).

Mehrabi, N. *et al.* (2021) 'A Survey on Bias and Fairness in Machine Learning', *ACM Computing Surveys*. Association for Computing Machinery. doi: 10.1145/3457607.

National Institute of Standards and Technology- US Department of Commerce (2022) 'Towards a Standard for Identifying and Managing Bias in Artificial Intelligence', *Natl. Inst. Stand. Technol. Spec. Publ*, 1270, p. 86. doi: 10.6028/NIST.SP.1270.

Oppenlaender, J. *et al.* (2020) 'Creativity on Paid Crowdsourcing Platforms', in *Conference on Human Factors in Computing Systems - Proceedings*. doi: 10.1145/3313831.3376677.

Schiffer, Z. (2021) 'Timnit Gebru was fired from Google — then the harassers arrived', *The Verge*, pp. 1–11. Available at: https://www.theverge.com/22309962/timnit-gebru-google-harassment-campaign-jeff-dean (Accessed: 11 April 2022).

The Irish Times (2021) 'Google Fires Second AI Ethics Leader as Dispute Over Research, Diversity Grows', *18 news*, p. 2. Available at: https://www.irishtimes.com/business/technology/google-fires-second-ai-ethics-leader-as-dispute-over-research-diversity-grows-1.4490768 (Accessed: 11 April 2022).

Tversky, A. and Kahneman, D. (1974) 'Judgment under Uncertainty: Heuristics and Biases', *Science*, 185(4157), pp. 1124–1131. Available at: http://www.jstor.org/stable/1738360.

Veale, M. and Binns, R. (2017) 'Fairer machine learning in the real world: Mitigating discrimination without collecting sensitive data', *Big Data and Society*, 4(2), pp. 1–17. doi: 10.1177/2053951717743530.

Algorithmic Censorship of Art: A Proposed Research Agenda

Piera Riccio
ELLIS Alicante
Alicante, Spain
piera@ellisalicante.org

Jose Luis Oliver
Architecture Department
Universidad de Alicante
Alicante, Spain
joseluis.oliver@ua.es

Francisco Escolano
Computer Science Department
Universidad de Alicante
Alicante, Spain
sco@ua.es

Nuria Oliver
ELLIS Alicante
Alicante, Spain
nuria@ellisalicante.org

Abstract

In the past decade, the application of Artificial Intelligence (AI) techniques to autonomously generate creative content or to support human creativity has gained interest from the scientific community. The generative models that have been proposed in the literature are changing the agency and dynamics of our art practices. A less explored area in the intersection of AI and creativity includes the indirect impact of AI on our creativity through content moderation algorithms on social media. Such algorithms tend to censor artistic pieces that display nudity, acting as inhibitors of human creativity. In this paper, we present a research agenda to tackle this challenge from a cultural and gender perspective, and we propose that a human and humanities-centered approach is necessary to develop AI systems that positively impact artistic practices.

Introduction

Social media platform adoption has grown exponentially in the past decade. Today, it is estimated that over 4.6 billion people in the world are active social media users[1]. For many of their users, these platforms have become the main source not only of social interactions, information and news (Walker and Matsa 2021), but also of their creative production and exposure to artistic content.

Artificial Intelligence (AI)-based algorithms are pervasive in social media platforms, to e.g. provide a personalized experience to their users, enable content search, target advertisements or automatically edit/filter images and videos. Content moderation[2] algorithms are a prominent example (Chen 2021). Protecting online users –particularly minors– from damaging content (e.g. violence, terrorism, hatred or pornography) is essential. Thus, most social media platforms publish community guidelines that define their content moderation policies. However, the immense volume of content posted and consumed daily on these platforms (e.g. over 90 million photos are posted on Instagram every day and more than 1 billion videos are viewed on TikTok daily) have led social media companies to heavily rely on AI-based algorithms for content moderation. Beyond inappropriate content, these algorithms tend to censor artistic pieces that display nudity –even when their intent is clearly non-sexual– constraining not only the freedom of expression of artists but also the cultural experiences of users.

Social media censorship concerns several aspects of our society and it is applied on a variety of artistic expressions. However, in this debate paper, we focus solely on the censorship of artistic nudity and we hypothesize that such censorship has a negative impact on the creative freedom of artists and on the broad diffusion of artistic content, eventually harming the users that they are trying to protect. As an example of such an impact, the Vienna museums created in 2021 an account on OnlyFans, an adult-only platform, after seeing their most famous artworks (by known artists, such as Schiele, Munch or Modigliani) repeatedly banned on Instagram, TikTok, and Facebook (Hunt 2021). The boundary between artistic nudes and pornography is highly debated among art theorists and sociologists (Vasilaki 2010; Patridge 2013; Eck 2001) and such an ambiguity is at the base of the cultural issue that we are addressing in our research.

In addition to the impact on the users of the platforms, several authors in the Computational Creativity (CC) community have argued that creativity needs to be situated and embodied in specific conditions to flourish (Saunders and Bown 2015; Guckelsberger et al. 2021). Considering social networks as a possible example of such an embodiment, censorship can have an impact on the inspiration for creative work not only for human authors but also for autonomous or co-creative systems that are immersed in this virtual environment, changing the nature of the artefacts that the system would be exposed to (Ritchie 2007). This negative impact of AI algorithms on social media contrasts the efforts of the scientific community, which in the past decade has shown great interest towards the development of AI algorithms that automatically generate art or assist humans in their creative processes. However, there is yet limited work in understanding the impact that such AI algorithms have on the cultural identity of our society. We believe that this subject deserves more attention from the computational creativity community. Hence, this short debate paper.

[1] https://datareportal.com/reports/digital-2022-global-overview-report

[2] Content moderation refers to the automatic prioritization, filtering, shadow-banning or censuring of content by means of AI-based algorithms.

Related Work

Social media is redefining the art world, from the marketing to the creation and curation of art. While these new dynamics and the democratization of art could be positive (Polaine, Street, and Paddington 2005), some authors claim that social media platforms have a negative impact on artistic production (James 2014) and creativity (Sharlow 2015). Manovich provides an overview of the connection between AI algorithms and the cultural ecosystems, emphasizing that the pervasiveness of AI algorithms is shaping our aesthetic decisions in creative media (Manovich 2018).

The algorithmic censorship of nudity on social media has been studied by several scholars, who have highlighted the disproportionate impact of such censorship on feminist artists (Faust 2017), and have explored the adopted artistic techniques to circumvent it (Olszanowski 2014). In recent years, artistic movements have emerged to publicly denounce the issue, such as *Don't Delete Art*[3] and *Artists Against Censorship*[4]. These initiatives and research related to this topic are of crucial importance to raise public awareness and to highlight the anthropological and sociological consequences of artistic censorship in social media. However, to the best of our knowledge, none of the existing initiatives address algorithmic censorship of art from a multidisciplinary perspective, including a technical analysis of the functioning of the content moderation algorithms.

AI-based algorithmic content moderation poses several societal challenges: first, such proprietary, machine learning-based algorithms are developed and maintained by private companies with clear economic incentives. Thus, their unprecedented power on defining our culture is exercised without any guarantee that it reflects the interests of society at large (Elkin-Koren 2020). Second, the automated decisions made by such algorithms are not always explainable and transparent, particularly if based on deep learning models. Third, algorithms are not foolproof and might not only make mistakes but also be fooled (Elkin-Koren 2020). Fourth, while historically controversial artistic content could be publicly discussed and debated, today artists have a limited ability to respond to censorship by social media platforms. Given the lack of transparency, it is hard to engage in a public debate if the reasons why certain content is banned are unknown. In contrast to related work, we propose a comprehensive research agenda on algorithmic censorship of art. Our objectives include an in-depth analysis of exemplary censored content, and the design of socio-technical solutions to mitigate such censorship.

AI and Art Censorship: A Historic Perspective

Nudity in the arts is historically considered *one of the defining aspects of mankind's creativity* (Deprez 2020). However, artistic nudes have been perceived, appreciated and accepted differently throughout history. Ancient Greeks conceived nudity as an expression of inner excellence, elevating humans from the realm of the flesh to the realm of Gods. In the Middle Ages, the same representations were perceived as obscene and sinful. In this period, classic paintings were covered and statues mutilated (Deprez 2019).

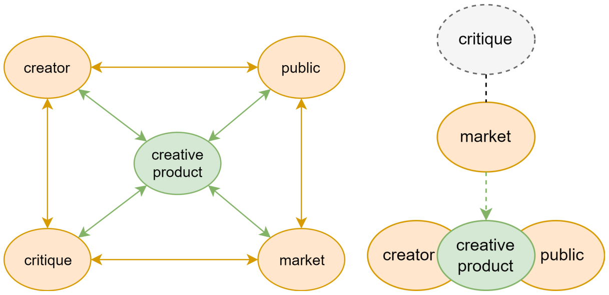


Figure 1: Synthetic sketch of the key elements within the creative ecosystem. Left, non-hierarchical arrangement among these elements before the advent of social media and AI; Right, transformation of the relationships in the context of AI algorithms used on social media.

The two aforementioned examples suggest that an understanding of the cultural context and ideals is necessary to embrace and appreciate the value of an artistic nude. Such context generally involves four key elements (critique/theory/context, market, public/observer and creators) to yield the creative product, as depicted in Figure 1. Historically (Figure 1, Left), these elements have been organized in a non-hierarchical structure, with connections among them. Depending on the artistic movement and the historic moment, one of these elements (for example the critique/theory) might have been more prominent that the rest in defining the environment for creativity (Montaner 1999). Studies in history of art identify and define the links and relations (depicted as arrows in the Figure) between the elements, and articulate a discourse about the artistic production from the perspective of different disciplines, including philosophy, morality, religion, politics, economics and aesthetics. Identifying the key elements and their relationships is crucial to develop a critical viewpoint of each creative framework, and to propose alternatives to it (Ramirez 1998). Today, these elements play new roles: the *public* is not simply a consumer, but it may become the product, i.e. the creation. Moreover, AI algorithms do not simply act as the *creators* (generating artistic content) but they can be, at the same time, the *critics* (deciding what is acceptable, and what is not) in a non-transparent way. We hypothesize that the ubiquity of opaque AI algorithms that impact the roles and links between the essential elements of the artistic creation environment hinders human creativity.

Art history is rich in examples of creative practices arisen from transgression and provocation towards existing ideals of morality. One such example is Michelangelo: despite working at the service of the Papacy, he depicted several nude figures in the iconic Sistine Chapel placing his masterpiece at risk of destruction (Vasari 1550). Unfortunately, disruptive artistic content might become an increasingly rarer phenomenon in our contemporary cultural environment (depicted in Figure 1, Right). AI algorithms, in fact, have the potential to not only influence one link in the diagram of the Figure 1, but simultaneously impact all the el-

[3] https://dontdelete.art/

[4] https://www.artistsagainstcensorship.com/

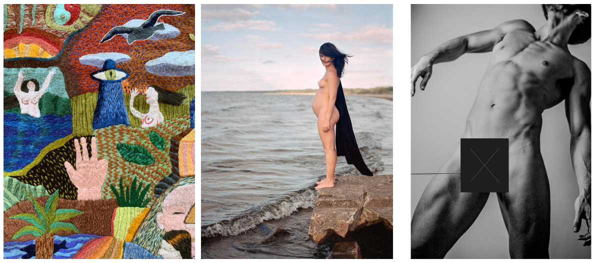


Figure 2: Three examples of censored images. Authors from left to right: Caroline Krabbe (collected through our survey), Adey (available on the *Artists Against Censorship* website), Udaentro (available on the *Don't Delete Art* website).

ements in the creative environment (Kulesz 2018). As a consequence, the traditional non-hierarchical structure morphs into a hierarchical organization where the *Market* lies on the top of the hierarchy, as the ultimate driver of the process, and therefore, as a fundamental agent in the creative decision-making process. Social media platforms are establishing a sort of monopoly to share content to the public. Algorithmic moderation on such platforms suffers from several important limitations: among others, we hypothesize that the utilized algorithms are unable to appreciate the value of an artwork or to understand the intent and context in which it is realized. As a consequence, social media leave no space for what is *blurred* (Kosko 1999) or *faint* (Vattimo 1988), drawing more defined –and yet invisible– lines between the acceptable and the unacceptable. In such a binary environment, breaking the rules is becoming harder, if not impossible.

AI and Art Censorship: Research Agenda

Given the importance of nudity in our artistic expression, we propose a research agenda on the topic of AI and algorithmic art censorship, articulated around four research questions.

RQ1: Pervasiveness of algorithmic censorship on social media

The first research question focuses on the pervasiveness of artistic nudity censorship on social medial platforms, its scope and characteristics.

Quantitative research in this domain is limited by the lack of representative, publicly available data, due to the proprietary nature of the social platforms and their content moderation algorithms. Hence, the first step in our research agenda entails reaching out to artist communities to collect a large corpus of censored artworks from social media. We are both establishing collaborations with relevant artists who have experienced censorship of their work and collecting additional examples of censored art through an online survey[5], which we launched in March of 2022.

The goal of this collection is to have a solid basis to shed light on the functioning of the content moderation algorithms and provide valuable feedback to artists as to why their content might have been shadow-banned or censored.

Preliminary analyses on the artworks that we have gathered to date reveal examples that depict female nudity with naivete (see first example in Figure 2), nudity without any sexual intent (see second example in Figure 2), or nudity that is already censored by the artist (see third example Figure 2). These pictures illustrate the extent of the issue that we plan to computationally analyze through the dataset.

RQ2: Human vs algorithmic censorship

The second research question aims to investigate the differences between the moral ideals embedded in today's content moderation algorithms and the human perception of art.

In 2021, the Facebook papers provided evidence that Meta maintains a *white list* of users[6] for which such content moderation rules do not apply. The inclusion in such a white list depends on the number of followers and popularity of a particular user. To highlight the market-driven decisions-making processes of content moderation algorithms, we plan to design and deploy a user study to collect ground truth on the appropriateness of the censored images (included in the dataset previously collected) when compared to other non-censored images displaying nudity. This research question aims to highlight the ability of people to recognize artistic intent in art and to show the existence of double standards on social media platforms. Given the broad reach of social media platforms across the planet, this user study will include a diverse set of participants from different cultural contexts to reflect the diversity of users in the platforms.

RQ3: Improved content moderation algorithms

Once we have a deeper understanding of the challenge, we plan to develop intent and context-aware content moderation algorithms that are able to distinguish artistic nudes from pornography.

Note that most of the social media platforms today do not explicitly ban artistic nudity in their community guidelines[7]. The discrepancy between the intent of the platforms and the actual censorship suggests that these algorithms are not yet refined enough to replace human moderators. In this regard, there is a need to develop content moderation algorithms that are intent and context-aware, combining different modalities (e.g. images and text) and leveraging inferred insights from the user study developed to address RQ2. Unfortunately, the existing ambiguity between artistic nudes and pornography is usually not taken into account by researchers developing algorithms for adult-content recognition (Wang et al. 2018; Chen 2021). We argue that an exploration of this issue could offer an opportunity in the field of Computational Creativity. In particular, the development of better content moderation algorithms for artistic nudity could leverage and improve the internal processes of evaluation in CC systems (Ventura 2017).

RQ4: Gender perspective

With RQ4, we address this topic with a gender perspective. The focus here is on studying the impact of such algorithms

[5] https://ellisalicante.org/censorship

[6] https://www.wsj.com/podcasts/the-journal Episode 1

[7] https://transparency.fb.com/it-it/policies/community-standards/adult-nudity-sexual-activity

on the cultural identity of women.

Throughout human history, women have been objectified in visual creative expressions (Barolsky 1999). While this pattern reappears with varied connotations in different historic time periods, the broad use of AI-based algorithms on social media could have unprecedented negative consequences for women. Remarkable feminist movements –such as *Free the Nipple* and *The Guerrilla Girls* (Pollen 2021)– have tried to raise social awareness about this issue.

In 1975, Mulvey (Mulvey 1975) identified the so called *male gaze* in Hollywood movies. This concept refers to a masculine heterosexual perception of women, who are depicted as objects of sexual desire, to satisfy what is known as *scopophilia* (i.e. the pleasure in looking). The concept of *male gaze* is still debated in today's visual culture. With the rise of social media, the *male gaze* has been argued to be stronger than it has ever been (Oliver 2017). We hypothesize that the censorship of female artistic nudes (and nipples, in particular) by AI algorithms has a role in this phenomenon. Today's AI algorithms on social media may be seen as socio-technical phenomena that automate culture through technology, perpetrating and possibly even amplifying human biases (Sezen 2020; Schroeder 2021). In particular, the censorship of female artistic nudity may be related to the conception of women as objects of pleasure. Because of this conception, female manifestations of nudity are frequently perceived as pornographic acts (Volkers 2020; Ibrahim 2017; Are 2021). This bias affects the freedom of expression of artists who are not conforming with the *male gaze* and that use female nudity to stand against the patriarchal sexualization of feminine bodies. Thus, we believe that the intersection between AI, social media, female nudity and art deserves to be further studied with a multi-disciplinary approach and a gender perspective.

Conclusion

In this paper we advocate for a research agenda focusing on the interplay between AI-based content moderation algorithms and art censorship on social media, and its implications on artistic production, creativity and the cultural identity of women. We have identified four broad research questions that would need to be addressed to fully understand such an interplay. These research questions (for example, the importance of having intent and context-aware content moderation algorithms) would need to be tackled *before* the widespread deployment of these technologies. Such a prior analysis would also entail interdisciplinary teams with experts from a variety of fields within the humanities and computer science (Crossick 2020). We emphasize the need to broaden the views of this research field, including both computing and non-computing disciplines (e.g. sociology, media studies, art history, anthropology) in the research agenda to develop technical solutions that are socially acceptable and responsible.

Author Contributions

P.R. and N.O. contributed to the proposal of the framework, proposal of the research questions, writing of the paper and its revision. J.L.O. contributed to the elements regarding the historic perspective. F.E. contributed to the paper's revision.

Acknowledgments

N.O. and P.R. are supported by a nominal grant received at the ELLIS Unit Alicante Foundation from the Regional Government of Valencia in Spain (Convenio Singular signed with Generalitat Valenciana, Conselleria d'Innovació, Universitats, Ciència i Societat Digital, Dirección General para el Avance de la Sociedad Digital). P.R. is also supported by a grant by the Banc Sabadell Foundation.

References

Are, C. 2021. The shadowban cycle: an autoethnography of pole dancing, nudity and censorship on instagram. *Feminist Media Studies* 1–18.

Barolsky, P. 1999. Looking at venus: A brief history of erotic art. *Arion: A Journal of Humanities and the Classics, (7):93–117.*

Chen, T. M. 2021. Automated content classification in social media platforms. In *Securing Social Networks in Cyberspace*. CRC Press. 53–71.

Crossick, G. 2020. From bridges to building sites: facilitating interdisciplinarity in the arts & humanities, last access: 19 may 2022.

Deprez, G. 2019. The destruction of nude images, last access: 31 may 2022.

Deprez, G. 2020. Cover up that bosom which i can't endure to look on, last access: 31 may 2022.

Eck, B. A. 2001. Nudity and framing: Classifying art, pornography, information, and ambiguity. *Sociological Forum* 16(4):603–632.

Elkin-Koren, N. 2020. Contesting algorithms: Restoring the public interest in content filtering by artificial intelligence. *Big Data & Society* 7(2):2053951720932296.

Faust, G. 2017. Hair, blood and the nipple. In *Digital Environments*. transcript Verlag. 159–170.

Guckelsberger, C.; Kantosalo, A.; Negrete-Yankelevich, S.; and Takala, T. 2021. Embodiment and computational creativity. *arXiv preprint arXiv:2107.00949*.

Hunt, E. 2021. Vienna museums open adult-only onlyfans account to display nudes, last access: 31 may 2022.

Ibrahim, Y. 2017. Facebook and the napalm girl: Reframing the iconic as pornographic. *Social Media + Society* 3(4):205630511774314.

James, P. 2014. 8 reasons why social media is decimating art and literature, last access: 31 may 2022.

Kosko, B. 1999. *The fuzzy future: from society and science to heaven in a chip*. Harmony.

Kulesz, O. 2018. Culture, platforms and machines: the impact of artificial intelligence on the diversity of cultural expressions. *Intergovernmental committee for the protection and promotion of the diversity of cultural expressions*.

Manovich, L. 2018. *AI aesthetics*. Strelka Press Moscow.

Montaner, J. 1999. Arquitectura y crítica. *Gustavo Gili.*

Mulvey, L. 1975. Visual pleasure and narrative cinema. *Screen* 16(3):6–18.

Oliver, K. 2017. The male gaze is more relevant, and more dangerous, than ever. *New Review of Film and Television Studies* 15(4):451–455.

Olszanowski, M. 2014. Feminist self-imaging and instagram: Tactics of circumventing sensorship. *Visual Communication Quarterly* 21(2):83–95.

Patridge, S. 2013. Exclusivism and evaluation: Art, erotica and pornography. In *Pornographic Art and the Aesthetics of Pornography*. Palgrave Macmillan UK. 43–57.

Polaine, A.; Street, S.; and Paddington, S. 2005. Lowbrow, high art: Why big fine art doesn't understand interactivity.

Pollen, A. 2021. Pubic hair, nudism and the censor: The story of the photographic battle to depict the naked body.

Ramirez, J. 1998. *Art History and critique: faults (and failures).* F. Cesar Manrique.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17:76–99.

Saunders, R., and Bown, O. 2015. Computational social creativity. *Artificial life* 21(3):366–378.

Schroeder, J. 2021. Reinscribing gender: social media, algorithms, bias. *Journal of Marketing Management* 37(3-4):376–378.

Sezen, D. 2020. Machine gaze on women: How everyday machine-vision-technologies see women in films. In *Female Agencies and Subjectivities in Film and Television*. Springer International Publishing. 271–293.

Sharlow, S. 2015. Death of an artist: How social media is ruining creativity, last access: 31 may 2022.

Vasari, G. 1550. *Le vite de' più eccellenti architetti, pittori, et scultori italiani, da Cimabue insino a' tempi nostri*.

Vasilaki, M. 2010. Why some pornography may be art. *Philosophy and Literature* 34(1):228–233.

Vattimo, G. 1988. *The End of Modernity: Nihilism and Hermeneutics in Post-Modern Culture*. Polity Press in Association with B. Blackwell.

Ventura, D. 2017. How to build a cc system. In *ICCC*, 253–260.

Volkers, R. 2020. Perverse media: How instagram limits the potential of feminist art, last access: 31 may 2022.

Walker, M., and Matsa, K. E. 2021. News consumption across social media in 2021, last access: 19 may 2022.

Wang, X.; Cheng, F.; Wang, S.; Sun, H.; Liu, G.; and Zhou, C. 2018. Adult image classification by a local-context aware network. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2989–2993.

Evaluation of Curriculum Learning Algorithms using Computational Creativity Inspired Metrics

Benjamin Fele, Jan Babič, Senja Pollak, Martin Žnidaršič
Jožef Stefan Institute
Jamova cesta 39
1000 Ljubljana
{benjamin.fele, jan.babic, senja.pollak, martin.znidarsic}@ijs.si

Abstract

Curriculum learning, especially in robotics, is an active research field aiming to devise algorithms that speed up knowledge acquisition by proposing sequences of tasks an agent should train on. We focus on curriculum generation in reinforcement learning, where various methods are currently compared based on the agent's performance in terms of rewards on a predefined distribution of target tasks. We want to extend this singular characterization of existing algorithms by introducing metrics inspired by notions from the field of computational creativity. Namely, we introduce surprise, novelty, interestingness, and typicality that quantify various aspects of tasks stochastically proposed by the curriculum learning algorithms for the learner to train on. We model proposed tasks with Gaussian mixture models which enable their probabilistic interpretation, and use Hellinger distances between distributions and training rewards in formulation of the proposed metrics. Results are presented for eight curriculum learning algorithms showcasing differences in prioritization of various aspects of task creation and statistically different mean metric values when comparing agent's best and worst training runs. The latter finding is not only useful for analysis of existing algorithms, but potentially also provides guidance for design of future curriculum learning methods.

Introduction

The idea of introducing tasks of increasing difficulty from the perspective of a student has a history in human learning (Oudeyer, Kaplan, and Hafner 2007) and teaching (Prideaux 2003), but similar ideas in machine learning have recently gained in popularity due to their ability to reduce the number of samples necessary for training or improvement in the final performance. Reinforcement learning algorithms often prohibit real-world applications due to inability to solve complex tasks from small number of agent's interactions with the environment, which is precisely why use of curricula has been popular in that domain.

While many practical advancements have been made in this field (Portelas et al. 2020; Gupta, Mukherjee, and Najjaran 2022), not many authors provide a theoretical analysis of their work. Some ideas about how curriculum learning works have been proposed by Bengio et al. (2009), suggesting that curriculum enables learning of smoothed convex functions first, allowing to reach a dominant (and possibly global) minimum before the loss function one is optimizing grows more complex. Kroemer, Niekum, and Konidaris (2021) also add that in practice, curriculum might enable the reinforcement learning agent to collect informative non-sparse rewards and thus aid training. Xu and Tewari (2021) on the other hand argue that in addition to above optimization benefits, statistical benefits are also important — curriculum algorithms can control the amount of variance the learning algorithm experiences, leading to faster convergence.

Analysis from the perspective of optimization theory provides an useful insight into inner-workings of the current curriculum learning algorithms, but are not the only way one should try to improve understanding of their characteristics. While theoretical analyses such as those in the previous paragraph are rare, we have not come across any works in the field of reinforcement learning pertaining evaluation and comparison of existing curriculum learning methods in terms other than accumulated agent's reward in the environment. Approaching evaluation by quantifying properties of generated curricula could provide insight into which approaches for curriculum generation work better and why, in addition to providing grounds for design of future algorithms.

Metrics introduced in this paper aim to fill the aforementioned gap. We do not analyse existing algorithms in terms of optimization theory, but instead evaluate them using metrics inspired by work from the field of computational creativity. Contributions of this paper are the following:

- We introduce surprise, novelty, typicality and interestingness for evaluation of curriculum learning algorithms in the scope of reinforcement learning. The metrics evaluate the tasks proposed by the algorithms from the perspective of the learning agent and are formulated using probabilistic measures ensuring their interpretability.

- We evaluate existing state-of-the-art curriculum learning approaches using our metrics. This provides insight into differences in prioritization of different aspects of curriculum generation and possible basis for future algorithms.

Related Work

In the scope of computational creativity, many frameworks and specific metrics have been proposed for evaluating creativity in the past. Computer generated artefacts that are evaluated using computational creativity metrics can be characterized across multiple dimensions and the exact formulations depend on their context. While the definitions differ across the field, there are existing metrics such as novelty (Boden 2004; Ritchie 2007; Elgammal and Saleh 2015; Canaan et al. 2018), surprise (Maher 2010; Grace and Maher 2014; Canaan et al. 2018), quality (Ritchie 2007), value (Boden 2004; Maher 2010; Elgammal and Saleh 2015; Canaan et al. 2018) or interestingness (Schmidhuber 2009; Canaan et al. 2018), among others, that each aim to provide means for evaluation of creative artefacts.

Boden (2004) introduced criteria describing new, surprising and valuable ideas as creative. She differentiated between what newness is to one person (P-creativity) or the whole human history (H-creativity), where the former guides the definition of our metrics. Many subsequent authors formulated some of their metrics based on her definition (Maher 2010; França et al. 2016). Wiggins (2006) bases his computational creativity formulation on Boden's work, but argues that the notion of surprise is redundant. Ritchie (2007) proposed assessment of creativity through novelty and quality in addition to matching the criteria of typicality. In our work, we take his idea of the latter as a measure of how well the artefact class in question is represented by the produced items. His definition of novelty is also useful for our formulation, since it describes produced item's dissimilarity to the already known artefacts. This definition of novelty is also similar to the one in Maher (2010).

While Wiggins (2006) doesn't differentiate between novelty and unexpectedness, some authors find it useful to separate the two. Unexpectedness, or surprise, can be defined as change of the generated artefacts compared to the recent past (Maher 2010; Grace and Maher 2014). Canaan et al. (2018) also differentiate between novelty and surprise, describing the former as a dissimilarity between collection of artefacts (distance-based novelty) and the latter as a measure of how much a generated sample differs from model's expectation. Surprise is also termed learning-based novelty in their work.

Schmidhuber (2009) describes a comprehensive theory of subjective beauty, interestingness, surprise and novelty providing a formulation of creativity. He starts by defining beauty from the perspective of an agent, describing it as a signal compressible to a large degree — in other words, finding it simple — and goes on to outline interestingness as a change in the perceived beauty. This is a template for our definition of interestingness as well. The formulation from Canaan et al. (2018) is also relevant, where the measure is defined as a specific value range of novelty. They stress the idea behind Wundt curve (Wundt 1874), explaining that too little or too much novelty might lead to uninteresting artefacts. Lastly, Reehuis et al. (2013) relate interestingness to learning progress, which is related to our formulation of this metric.

In the scope of practical applications, França et al. (2016) and Varshney et al. (2019) implement novelty as a Bayesian surprise. Bayesian surprise is large whenever the impact of new data on the prior distribution is large (Franceschelli and Musolesi 2021). Novelty can also be based on simple distance measures like in Morris et al. (2012) and Maher (2010). Quality and value can on the other hand be evaluated using artificial neural networks (Morris et al. 2012), distance between nodes in a graph (Elgammal and Saleh 2015) or associations of an artwork with its description (Norton, Heath, and Ventura 2010).

Curriculum Generation in Reinforcement Learning

The use case of our proposed metrics is in the scope of automatic curriculum generation. While curricula can be used in many machine learning fields, we focus specifically on its use in reinforcement learning. Before continuing to our proposed metrics, we therefore introduce the framework within which they are utilized.

Reinforcement Learning

Reinforcement learning (RL) is defined in the scope of Markov Decision Process (MDP) denoted by a tuple $M_{RL} = (\mathcal{S}_{RL}, \mathcal{A}_{RL}, R_{RL}, \mathcal{P}_{RL}, \gamma_{RL})$, with $\mathcal{S}_{RL}$ being state space, $\mathcal{A}_{RL}$ action space, R_{RL} reward function, $\mathcal{P}_{RL}$ the state transition probabilities (i.e. environment dynamics) and γ_{RL} the discount factor prioritizing long-term planning (Sutton and Barto 2018). In a reinforcement learning algorithm, an agent performs an action $a \in \mathcal{A}$ upon experiencing state $s_t \in \mathcal{S}$, receiving a new state s_{t+1} and a reward $r = R_{RL}(s_t, a_t, s_{t+1})$ in return. The goal is to find the optimal policy π^* that maximizes the expected reward for every possible state (Sutton and Barto 2018). In addition to searching for the optimal policy π^*, solutions to auxiliary objectives have to be found: state value function $V_\pi(s_t)$ and state-action value function $Q_\pi(s_t, a_t)$, that estimate achieved reward under current policy until the end of an episode. These functions are used to search for the optimal policy, for example by greedy selection of actions leading to highest expected rewards, or as a guiding signal for policy gradient algorithms.

Curriculum Learning

As implied in the previous paragraph the agent interacts with the environment by performing actions. However, in order to learn the optimal policy, it has to randomly explore its actions and the state space. Curriculum learning (CL) aims to narrow down this exploration space by bounding it to subtasks that an agent should master first (Narvekar et al. 2020). Automatic curriculum learning, to which the methods evaluated in this paper belong, generates curricula algorithmically depending on agent's progress. The problem of finding the sequence of tasks that lead to fastest learning and best performance can be seen as finding the optimal policy for a secondary MDP $M_{CL} = (\mathcal{S}_{CL}, \mathcal{A}_{CL}, R_{CL}, \mathcal{P}_{CL}, \gamma_{CL})$, where $\mathcal{S}_{CL}$ are representations of agent's policy or knowledge, $\mathcal{A}_{CL}$ control task difficulty, R_{CL} is the reward function, e.g. the accumulated agent's reward on a target task distribution P_{target} (Narvekar and Stone 2019). The target task distribution P_{target} entails all possible subtasks that we want our agent to generalize over. Within M_{CL}, γ_{CL} prioritizes immediate versus long-term consequences of the chosen tasks.

Above definition describes the problem to be solved in curriculum learning, but says little about actual implementations. In practice, exhaustively searching for an optimal curriculum is often intractable and can take longer than training the agent without curriculum to achieve the same performance (Narvekar and Stone 2019). Researchers thus utilize various heuristics to simplify search for good solutions. In this regard, the notion of learning progress has been a valuable measure for choosing suitable subtasks to train from in the past (Baranes and Oudeyer 2010; Moulin-Frier, Nguyen, and Oudeyer 2014; Portelas et al. 2020; Colas et al. 2019). Learning progress is defined as the (absolute) difference of collected rewards over time in some part of the task space $\mathcal{T}_{env}$. For example, RIAC (Baranes and Oudeyer 2010), Covar-GMM (Moulin-Frier, Nguyen, and Oudeyer 2014) and ALP-GMM (Portelas et al. 2020) all use such criteria in their frameworks, varying how they split the task space — across one (RIAC) or multiple (Covar-GMM and ALP-GMM) dimensions — or how they implement learning progress — through covariance between rewards and time (Covar-GMM) or absolute difference between new and old rewards (RIAC and ALP-GMM). However, not all methods rely on learning progress for selection of task parameters. ADR (Akkaya et al. 2019) expands the distribution from which the task parameters are sampled by some δ depending on whether the agent achieved sufficient performance on current subtasks. Klink et al. (2020) on the other hand propose Self-Paced curriculum learning algorithm, taking a probabilistic approach to gradually expand the task parameter distribution towards target one by weighting the loss term of their algorithm according to the agents performance. With the increasing popularity of generative adversarial networks, GoalGAN (Florensa et al. 2018) and Setter-Solver (Racaniere et al. 2019) algorithms take advantage of competition between subtask proposition and subtask suitability estimation modules. We evaluate most of the approaches outlined in this paragraph using our proposed metrics.

A common trait of the previously outlined algorithms is the mechanism by which they generate a curriculum. They all control the environment difficulty through subtask selection. This is performed by sampling task parameters for environment initialization from the task space $\mathcal{T}_{env}$. That being said, other ways of generating curricula also exist (Schaal 2006; Andrychowicz et al. 2017; Zhou et al. 2019; Zhang, Abbeel, and Pinto 2020). The way task parameters are sampled during evaluation is determined by P_{target} which is usually normally (Klink et al. 2020) or uniformly (Portelas et al. 2020) distributed.

Framing in the Context of Computational Creativity

In order to bridge the gap between curriculum learning and computational creativity, it is useful to frame the former in the context of the latter. For this reason, we turn to Wiggins (2006), who formalized Boden's (2004) model of computational creativity. He defines it as a tuple $(\mathcal{U}, \mathcal{L}, [[\cdot]], \langle\langle\cdot,\cdot,\cdot\rangle\rangle, \mathcal{R}, \mathcal{T}, \mathcal{E})$, where $\mathcal{U}$ is the universe of all concepts, $\mathcal{L}$ is the language used for expressing acceptable concept space $\mathcal{R}$, concept search algorithm $\mathcal{T}$ and concept evaluation function $\mathcal{E}$. Functions $[[\cdot]]$ and $\langle\langle\cdot,\cdot,\cdot\rangle\rangle$ are functions that apply ruleset $\mathcal{R}$ and generate concepts using $\mathcal{R}$, $\mathcal{T}$ and $\mathcal{E}$, respectively.

Within the above formulation, space of all subtasks in a curriculum learning framework can be framed as $\mathcal{U}$, while $\mathcal{R}$ bounds this space to the target task distribution or subtasks achievable by the agent. $\mathcal{T}$ can then be an algorithm modelling the curriculum, and $\mathcal{E}$ is the curriculum evaluation metric. The latter is usually the agent's reward achieved during testing, but can also be learning progress or criteria of validity, feasibility and coverage (Racaniere et al. 2019). Metrics, proposed in the next section can also belong to the set $\mathcal{E}$. Lastly, function $[[\cdot]]$ filters tasks not conforming to $\mathcal{R}$ thus bounding possible subtasks to some subspace, and function $\langle\langle\cdot,\cdot,\cdot\rangle\rangle$ generates the actual curriculum. Language $\mathcal{L}$ is in our case a set of real numbers describing specific subtasks.

Proposed Metrics

We define four metrics for evaluation of curriculum generation algorithms in relation to the agents learning to perform a specified task. For three out of four metrics we use Hellinger distance (Hellinger 1909) between various distributions to capture the changing nature of the underlying task sampling process. The distance is defined as:

$$H(f,g) = \sqrt{\frac{1}{2}\int \left(\sqrt{f(x)} - \sqrt{g(x)}\right)^2 dx}, \tag{1}$$

where f and g are probability density functions corresponding to the two distributions we want to measure the distance between. Hellinger distance has some advantageous properties compared to other probability-based distance measures: it is for example symmetric and bound to an interval $0 \leq H(f,g) \leq 1$, which simplifies the comparison and interpretation of our results. We split the raw task parameters into windows of fixed sizes as described in the Task Parameter Prepocessing section, forming the basis for computation of the following metrics.

Surprise, novelty and typicality are all based on the aforementioned distance measure. The central idea behind surprise and novelty revolves around measuring short- and long term changes of proposed tasks, while typicality aims to capture their overlap in regards to the target task distribution. Our formulation of surprise and novelty is largely inspired by Maher (2010) by taking an idea of surprise as a measure of change in the distribution expectation compared to the recent past, and novelty describing the difference between the new and already existing data.

We define *surprise* as:

$$S_t = H(P_t, P_{t-1}), \tag{2}$$

and *novelty* as:

$$N_t = \frac{1}{t}\sum_{k=0}^{t-1} H(P_t, P_k), \tag{3}$$

where P above denotes probability density functions of distributions fitted at specified time-points. Surprise captures changes between two sequential distributions at time t and $t-1$, while for novelty, we compute mean changes between tasks in t-th window compared to the distributions of all windows prior to it. At $t = 1$, the definition of novelty is equal to surprise, but they measure a different quantity as t grows larger. Note, that above definitions are only sensible for $t > 0$.

For *typicality*, we turn to Ritchie (2007), who formulated it as a measurement of the extent the produced item is an example of the artefact class in question. With slight abuse of this notion, we take the task distribution P_{target} that delimits the scope of problems we want our agent to be able to solve and measure its distance from distribution of tasks P_t:

$$T_t = 1 - H(P_t, P_{target}). \tag{4}$$

This formulation yields high typicality when the distance between some task parameter and target distribution is low. In our experiments, task parameters underlying the distribution P_t are bounded by P_{target}. The latter is in our case uniformly distributed and in such settings this metric measures what might be better denoted as *coverage*. However, in general, P_{target} could follow any other distribution, so we keep referring to it as *typicality* in this paper.

Lastly, the *interestingness* is defined according to the reasoning behind such metric in Schmidhuber (2009). This is the only metric that does not look at the proposed task parameter distributions themselves, but is instead computed from agent's received rewards during training. It measures the change in agent's collected rewards in a particular period of training, assuming that the underlying task parameter distribution is not exhibiting sudden changes. This way, our proposed measure is related to the change of simplicity from the perspective of an agent, as proposed by Schmidhuber (2009).

A naive approach entailing simple subtraction of rewards in the first and second halves of the window has problems with taking into account changing number signs and is not bounded to any specific interval. In order to remedy this issue we use cumulative density function to bound the interestingness values. First, let's assume that the rewards collected in a specified time period are normally distributed. Cumulative density function Φ is then defined as (Walck 2007):

$$\Phi\left(\frac{x-\mu_t}{\sigma_t}\right) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-\frac{x^2}{2}}, \tag{5}$$

where x is an input variable, and μ_t and σ_t are in our case the mean and standard deviation of the rewards collected during training within a particular period t. After splitting that period in half, we obtain $\mu_{1/2}$ and $\mu_{2/2}$ corresponding to respective means of the two halves. These statistics are finally used to compute interestingness:

$$I_t = \phi\left(\frac{\mu_{2/2}-\mu_t}{\sigma_t}\right) - \phi\left(\frac{\mu_{1/2}-\mu_t}{\sigma_t}\right). \tag{6}$$

The use of cumulative density function ensures that each of the above terms is bound to an interval $[0, 1]$ and supports the notion that large deviations of $\mu_{1/2}$ and $\mu_{2/2}$ from the mean μ_t will result in larger absolute value of interestingness. When $\mu_{2/2} > \mu_{1/2}$ the interestingness will be positive, while in the other case its value will be negative, bounding the metric to an interval $[-1, 1]$.

Experiments

The metrics proposed above are used for evaluation of results of various curriculum learning algorithms. This section describes the benchmark setup from which the results were obtained, in addition to necessary prepossessing steps enabling treatment of task parameters guiding agent's training in a probabilistic manner.

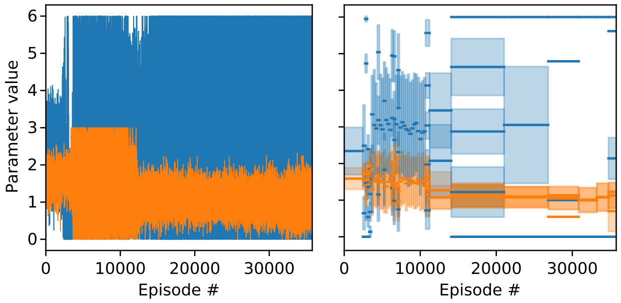
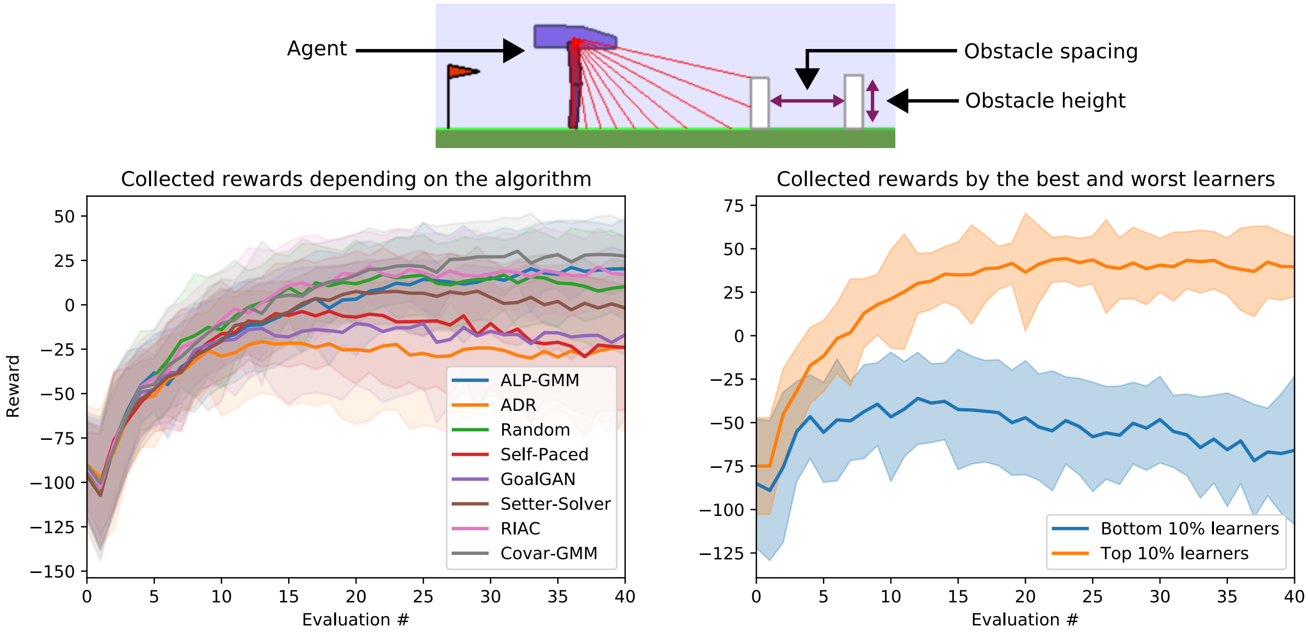


Figure 1: An example of raw task parameters (left), and GMM means and standard deviations after preprocessing (right) for every episode of training. Values of obstacle spacings are in blue and values for their heights are in orange.

Task Parameter Prepocessing

As outlined in the Curriculum Learning section, curriculum learning methods evaluated using proposed metrics operate by proposing task parameters used for environment initialization and thus controlling its difficulty. As the agent progresses, various approaches estimate distributions from which these initialization parameters are sampled. Some algorithms sample from uniform (Akkaya et al. 2019) or (multiple) Gaussian (Moulin-Frier, Nguyen, and Oudeyer 2014; Portelas et al. 2020; Klink et al. 2020) distributions, while others might use a more complex sampling scheme (Florensa et al. 2018; Racaniere et al. 2019). To enable computation of our metrics, we model the distributions of sampled task parameters during agent's learning using Gaussian mixture models (GMMs). Without knowledge of the real underlying distributions, this provides a reasonable estimate and a basis for probabilistic interpretation of the task parameter sampling process. This way, the process is not oversimplified as it would be if the normal distribution was assumed across all evaluated task distributions. This allows our use of Hellinger distance as a measure of change between two algorithms and subsequent analysis.

To capture the underlying distribution from which task parameters are sampled at different time-points during training, we split them into smaller windows w_t ($t \in [0, 40]$).



Collected rewards depending on the algorithm

Collected rewards by the best and worst learners

Figure 2: Top: an image of the agent in the environment and the two parameters controlling its difficulty. Bottom left: rewards collected during periodic testing while training. Bottom right: testing rewards for the highest and lowest $10\ \%$ of learners.

An example of raw task parameter data is seen in Figure 1 (left) and respective splits with fitted GMMs are seen in Figure 1 (right). Figures show the evolution of two task parameters used for controlling the difficulty of the agent moving through the environment. These two parameters control obstacle heights and spacings and are visualized in Figure 2 (top).

Each window size in Figure 1 corresponds to the time it took an agent to perform 500000 steps, resulting in a varying number of episodes in each window. Since the task parameters are sampled per episode, this also results in windows apparently varying in size in Figure 1 (right). GMMs with up to 5 components are fitted on the task parameters in each window and the one with the largest Akaike information criterion (Sakamoto, Ishiguro, and Kitagawa 1986) is kept for further analysis. This is a metric that quantifies how well the GMMs fit the underlying data.

Experimental Setup

TeachMyAgent benchmark (Romac et al. 2021) provides a framework for evaluation of various curriculum and reinforcement learning algorithms in two environments in multiple training configurations. It tests 7 curriculum learning algorithms in addition to the random baseline. The curriculum generation algorithms available in the TeachMyAgent benchmark and also used for evaluation of our metrics are ALP-GMM (Portelas et al. 2020), ADR (Akkaya et al. 2019), Self-Paced (Klink et al. 2020), GoalGAN (Florensa et al. 2018), Setter-Solver (Racaniere et al. 2019), RIAC (Baranes and Oudeyer 2010) and Covar-GMM (Moulin-Frier, Nguyen, and Oudeyer 2014). They are already briefly outlined in Curriculum Learning section. We take their results from the StumpTracks environment (Portelas et al. 2020), which is illustrated in the Figure 2 (top). The environment consists of an agent being tasked to learn to walk in environments with varying obstacle heights and spacings.

TeachMyAgent authors also introduced multiple agent embodiments and training configurations. We evaluate our metrics on results using a bipedal walker with an uniform target task distribution P_{target} bounded to the interval $[0, 3]$ for obstacle height and $[0, 6]$ for obstacle spacing. Data used in our analysis entails task parameters used for initialization of environments for each episode and respective accumulated rewards in addition to cumulative rewards obtained during each testing period.

As mentioned before, the agent's performance is periodically tested in the environment initialized with the parameters from the target task distribution P_{target}. Figure 2 shows the average performance during testing phases of the learning agent depending on curriculum learning algorithm used (bottom left) or its overall performance (bottom right). This illustrates that the differences between each curriculum generation method are in this case relatively small, but best and worst performances vary more substantially. Notice, how some algorithms in Figure 2 (bottom left) perform worse than the random baseline; this is in line with the results presented by Romac et al. (2021).

Each curriculum learning method in Figure 2 and the Results section was evaluated on results obtained from 64 experiment runs each using a different random seed. The points at which the agent is tested serve for splitting the task initialization parameters proposed by the curriculum learning algorithms into smaller windows. The best and worst

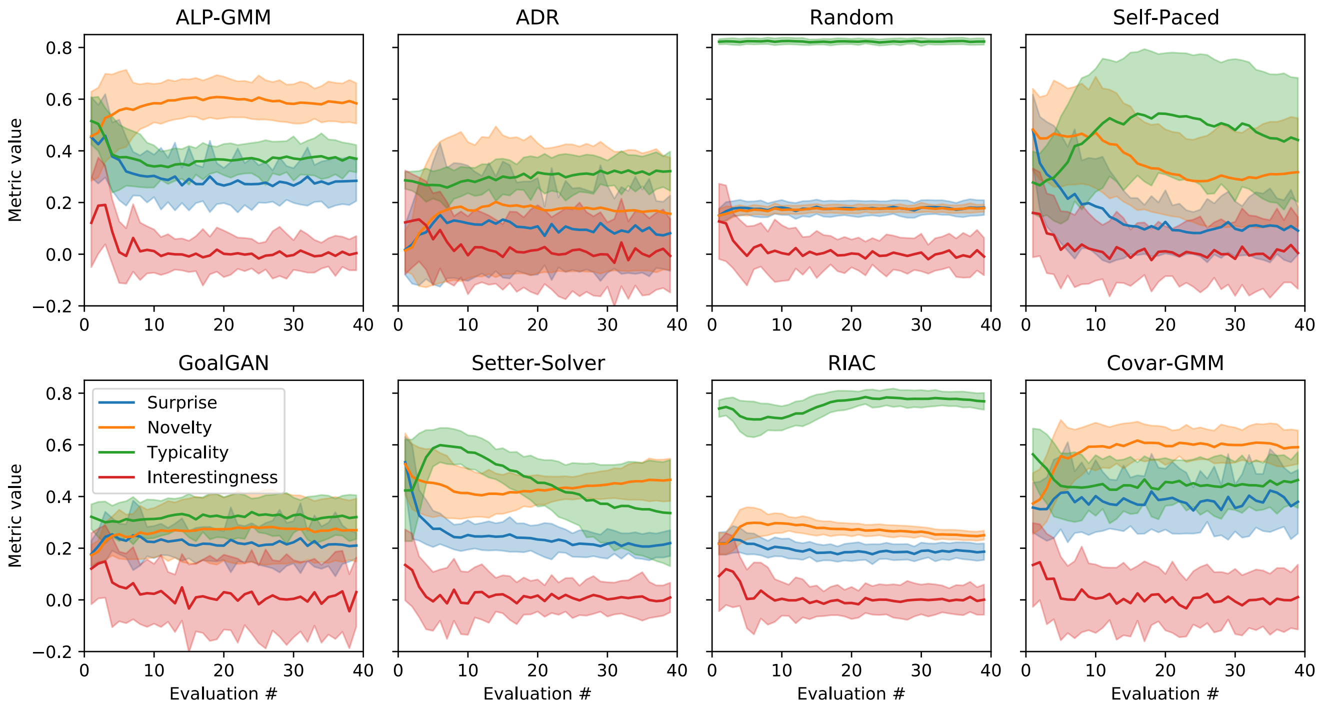


Figure 3: Metric values through agent training for all tested algorithms. It can be seen that every algorithm yields a different progression of metrics.

10 % of experiment runs used for visualization in Figures 2 and 5 are obtained by taking the distribution of mean collected test rewards and extracting bottom and top performers. This results in 52 samples in each group. We use Welch's t-test with $\alpha = 0.05$ with Bonferroni correction to evaluate statistically significant differences in metric values. For computation of Hellinger distances we use Monte-Carlo integration with 1000 samples, yielding a reasonably low error.

Results

Our metrics can be evaluated from two viewpoints presented in the following subsections. One perspective takes evolution of the metric values over various curriculum learning algorithms and thus provides the means for their comparison, while the other concentrates on agent's performance regardless of the underlying curriculum generator, highlighting changes between better and worse training runs.

Comparison of Curriculum Learning Algorithms

Figure 3 shows resulting values of our metrics depending on the algorithm they were evaluated on. At first glance, there are considerable differences between algorithms giving each of them a particular *silhouette*. Metrics usually vary the most at the beginning of training, and later stabilize at some value. We speculate their fast convergence is a consequence of the fact that over time the subset of tasks to be mastered gets smaller which is mirrored in the differences in their distributions. Self-Paced and Setter-Solver curriculum generation algorithms are an exception to this rule where novelty and typicality metrics don't converge like described. Results obtained from training with random curriculum show metric values when task parameters are uniformly distributed throughout training. Typicality, measuring the similarity between proposed and target task distributions, is in this case consistently the highest, but not equal to 1 due to inability of Gaussian mixture models to accurately capture uniform target task distribution. This is also the reason why the typicality for random curriculum is consistently high — there, the subtasks are sampled uniformly from the target task distribution.

Regardless of the algorithm they are evaluating, novelty and surprise hold similar values at the beginning of training, and grow more dissimilar later. Since the distribution underlying random curriculum generation doesn't change in the course of training, its surprise and novelty hold consistent values in the lower end of the spectrum. Interestingly, as seen in the Figure 4, ADR holds similar or lower values in this metrics and Self-Paced curriculum learning algorithm starts with a relatively high value of surprise but converges to around 0.1. In general, RIAC seems to hold values close to the ones obtained by the random curriculum and also leads to similar agent's performance as seen in the Figure 2. ALP-GMM and Covar-GMM show the highest novelty of the proposed subtasks, and generally hold similar metric values.

Comparing standard deviations in Figure 3, some algorithms (ALP-GMM, Setter-Solver, RIAC and randomly generated curriculum) exhibit smaller diversities of the com-

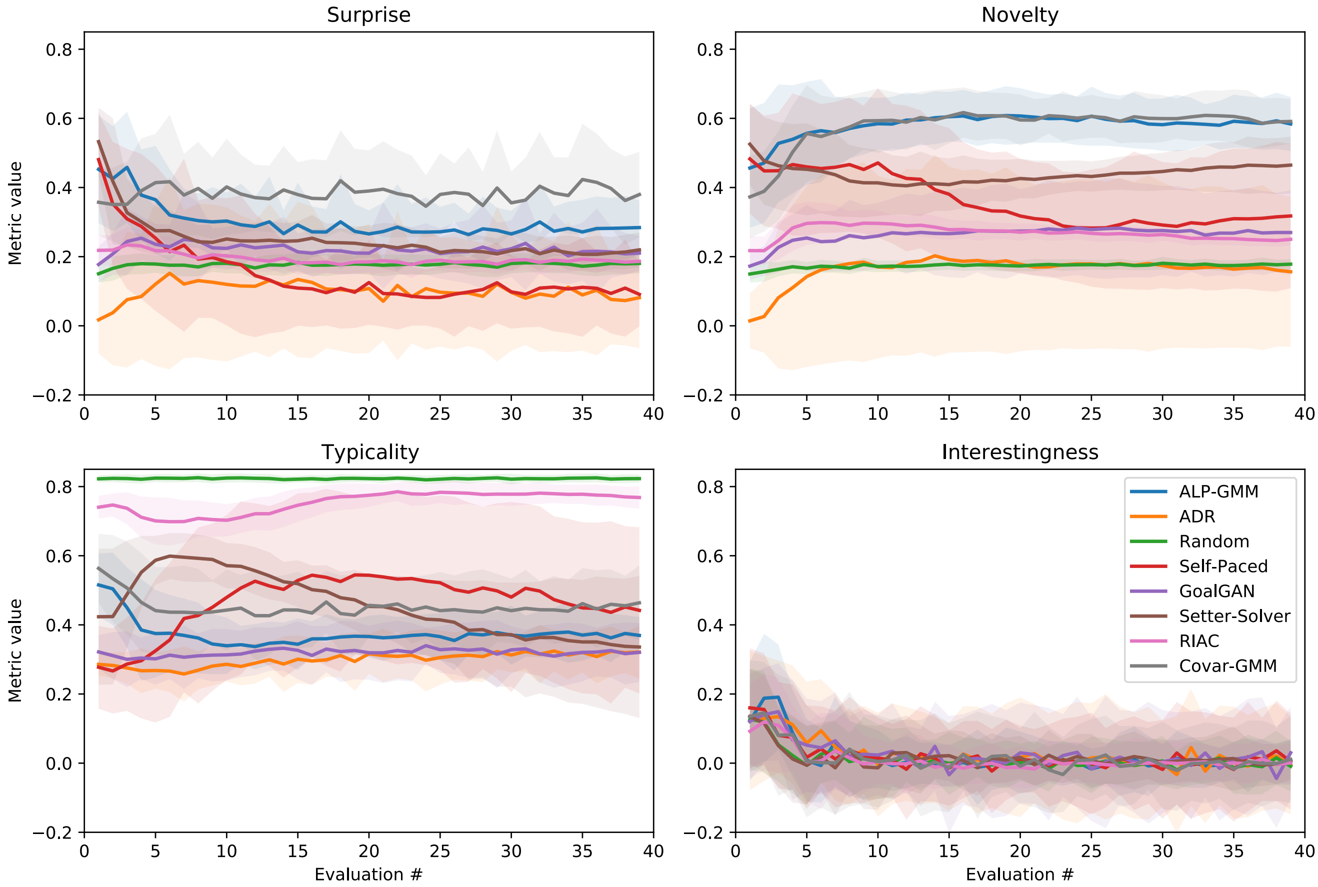


Figure 4: Results measuring tested algorithms shown by the metrics used. It can be seen that interestingness has the least variation across all curriculum learning algorithms.

puted measures, while others are more variable across experiment runs. Overall, interestingness seems to change between evaluated algorithms the least, which is clearly seen in Figure 4. From this perspective, it is not as useful for evaluation of curriculum learning algorithms as the other metrics. Most variability in this metric comes at the beginning of learning, when the agent's knowledge consistently starts improving. When comparing metrics other than interestingness between each other, it can be seen that they take distinctly different values and are in this sense not redundant.

Best- and Worst-performing Experimental Runs

As the differences between performances depending on the chosen algorithm are relatively small, this is not a suitable viewpoint for evaluation of curricula characterized by our metrics in regards to agent's performance. As shown in Figure 5, all metrics except interestingness consistently exhibit statistically different means when evaluated in regards to the best and worst training runs: $2.017 \times 10^{-16} < p < 4.306 \times 10^{-4}$ for surprise, $2.36 \times 10^{-15} < p < 4.728 \times 10^{-4}$ for novelty and $6.272 \times 10^{-11} < p < 4.584 \times 10^{-4}$ for typicality. Interestingness is not consistently significant with p-values $1.693 \times 10^{-2} < p < 36.056$. Surprise, novelty and typicality exhibit higher values with better learners. Visible trend in evolution of surprise and typicality is not obvious, but it is more clearly present when measuring novelty. Namely, with better performing learners it starts at a lower value and stabilizes around 0.45, a trend not present with evaluation of the bottom $10\ \%$ of the learners.

The lack of perceived trend might on one hand come from large diversity of surprise, novelty and typicality across training runs, also visible in relatively large variability of these metrics across algorithms in Figure 4. On the other hand, the metrics in general seem to stabilize at some value, which could also provide an explanation for the lack of trends on the graphs.

Interestingness is an exception in regards to patterns described in the above paragraphs. As was already seen in Figure 4, the metric shows low variability between curriculum generation algorithms, and is also not consistently significantly different in Figure 5. At the beginning of training, the interestingness stays similar across the two groups, but later settles at lower values for both groups. The values of better performing learners turn out to stabilize lower and have smaller standard deviation compared to the worse performing ones.

Since interestingness measures agent's progress during training, larger values correspond to faster learning of the

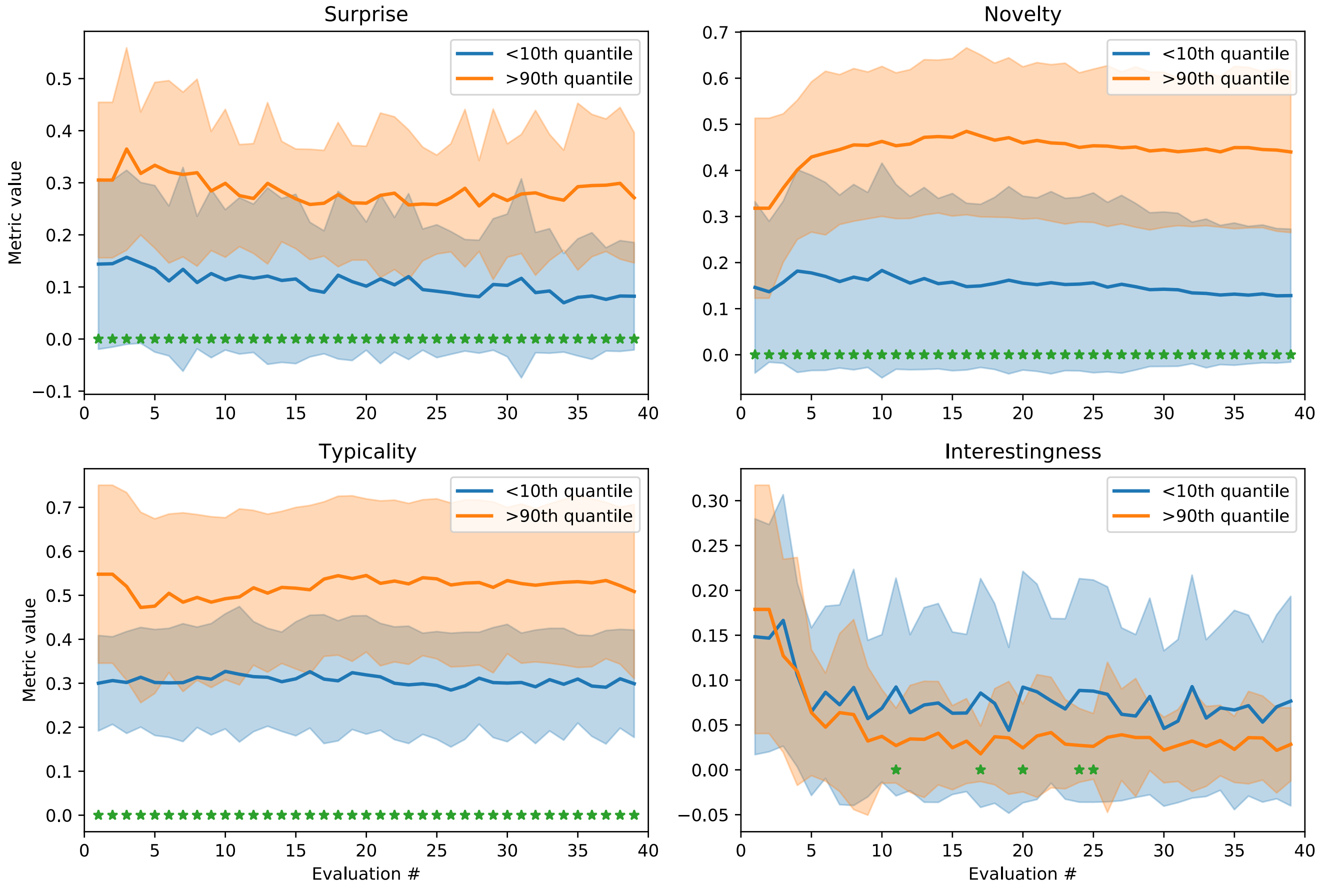


Figure 5: Mean metric values and standard deviations during agent's training for best and worst 10 % performing learners. Stars at the bottom of plots denote statistically significant changes.

tasks given in a particular time-period. This shows that fast improvements on specific subtasks during training don't translate into better performing agents on the target task distribution used during evaluation — our findings show that the opposite is true. This would imply that the tasks, even though they are labeled as more interesting by our metric, are perhaps less relevant for agent's progress on the target task.

Conclusion

This paper formulates metrics inspired by notions from the field of computational creativity and uses them for evaluation of curriculum learning algorithms. Results show that our metrics exhibit informative characteristics from two points of view: (i) as the means to differentiate and characterize curricula generated by different algorithms and (ii) distinguish more successful training runs from the less successful ones.

The differences between best and worst performing learners highlight that higher values of surprise, novelty and typicality, and lower values of interestingness, are generally beneficial for learning and its overall performance. Higher surprise signifies that more sudden changes in task distributions are beneficial, which also holds for coverage of the target task distribution implied by results for novelty and typicality. Interestingness results are less interpretative in our case, but suggest that proposing tasks resulting in larger values of this metric doesn't translate into better overall performance.

The property of interestingness to unsuccessfully capture what it was intended for is one of the shortcomings of our work. Furthermore, more tests should be conducted to determine how the proposed metrics correlate with actual underlying tasks that we are trying to model; notice how our approach is not concerned with mechanisms behind curriculum generation, subtask order or learner choice. Some of these issues could be remedied by conducting more detailed analysis using all results provided by the TeachMyAgent benchmark, and also obtaining some of our own.

Shortcomings aside, we want to stress that the results still provide a good starting point for design of future curriculum learning algorithm; for example, the selection of subtasks could be guided by balancing the values of the proposed metrics. This way, the utilization of our metrics could serve as a guiding criteria for determination of suitable subtasks that the agent should train on and contribute to the future of algorithmic curriculum generation.

Author Contributions

B. Fele carried out the analysis and wrote the core of the paper. B. Fele, S. Pollak and M. Žnidaršič conceptualized the metrics presented in this paper. B. Fele, S. Pollak, M. Žnidaršič and J. Babič worked on amending and correcting the text of the paper.

Acknowledgments

This work was supported by the Slovenian Research Agency (research core funding no. P2-0076 and P2-0103).

References

Akkaya, I.; Andrychowicz, M.; Chociej, M.; Litwin, M.; McGrew, B.; Petron, A.; Paino, A.; Plappert, M.; Powell, G.; Ribas, R.; et al. 2019. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*.

Andrychowicz, M.; Wolski, F.; Ray, A.; Schneider, J.; Fong, R.; Welinder, P.; McGrew, B.; Tobin, J.; Abbeel, P.; and Zaremba, W. 2017. Hindsight experience replay. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 5055–5065.

Baranes, A., and Oudeyer, P.-Y. 2010. Intrinsically motivated goal exploration for active motor learning in robots: A case study. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1766–1773. IEEE.

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, 41–48.

Boden, M. A. 2004. *The creative mind: Myths and mechanisms*. Routledge.

Canaan, R.; Menzel, S.; Togelius, J.; and Nealen, A. 2018. Towards game-based metrics for computational co-creativity. In *2018 IEEE conference on Computational Intelligence and Games (CIG)*, 1–8. IEEE.

Colas, C.; Fournier, P.; Chetouani, M.; Sigaud, O.; and Oudeyer, P.-Y. 2019. Curious: intrinsically motivated modular multi-goal reinforcement learning. In *International conference on machine learning*, 1331–1340. PMLR.

Elgammal, A., and Saleh, B. 2015. Quantifying creativity in art networks. In *Proceedings of the Sixth International Conference on Computational Creativity June*, 39.

Florensa, C.; Held, D.; Geng, X.; and Abbeel, P. 2018. Automatic goal generation for reinforcement learning agents. In *International conference on machine learning*, 1515–1528. PMLR.

França, C.; Góes, L. F. W.; Amorim, A.; Rocha, R.; and Da Silva, A. R. 2016. Regent-dependent creativity: A domain independent metric for the assessment of creative artifacts. In *Proceedings of the Seventh International Conference on Computational Creativity*, 68–75. Citeseer.

Franceschelli, G., and Musolesi, M. 2021. Creativity and machine learning: A survey. *arXiv preprint arXiv:2104.02726*.

Grace, K., and Maher, M. L. 2014. What to expect when you're expecting: The role of unexpectedness in computationally evaluating creativity. In *ICCC*, 120–128. Ljubljana.

Gupta, K.; Mukherjee, D.; and Najjaran, H. 2022. Extending the capabilities of reinforcement learning through curriculum: A review of methods and applications. *SN Computer Science* 3(1):1–18.

Hellinger, E. 1909. Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. *Journal für die reine und angewandte Mathematik* 1909(136):210–271.

Klink, P.; D' Eramo, C.; Peters, J. R.; and Pajarinen, J. 2020. Self-paced deep reinforcement learning. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 9216–9227. Curran Associates, Inc.

Kroemer, O.; Niekum, S.; and Konidaris, G. 2021. A review of robot learning for manipulation: Challenges, representations, and algorithms. *J. Mach. Learn. Res.* 22:30–1.

Maher, M. L. 2010. Evaluating creativity in humans, computers, and collectively intelligent systems. In *Proceedings of the 1st DESIRE Network Conference on Creativity and Innovation in Design*, 22–28. Citeseer.

Morris, R. G.; Burton, S. H.; Bodily, P.; and Ventura, D. 2012. Soup over bean of pure joy: Culinary ruminations of an artificial chef. In *ICCC*, 119–125.

Moulin-Frier, C.; Nguyen, S. M.; and Oudeyer, P.-Y. 2014. Self-organization of early vocal development in infants and machines: the role of intrinsic motivation. *Frontiers in psychology* 4:1006.

Narvekar, S., and Stone, P. 2019. Learning curriculum policies for reinforcement learning. In *Proceedings of the 18th International Conference on Autonomous Agents and Multi-Agent Systems*, 25–33.

Narvekar, S.; Peng, B.; Leonetti, M.; Sinapov, J.; Taylor, M. E.; and Stone, P. 2020. Curriculum learning for reinforcement learning domains: A framework and survey. *arXiv preprint arXiv:2003.04960*.

Norton, D.; Heath, D.; and Ventura, D. 2010. Establishing appreciation in a creative system. In *ICCC*, 26–35.

Oudeyer, P.-Y.; Kaplan, F.; and Hafner, V. V. 2007. Intrinsic motivation systems for autonomous mental development. *IEEE transactions on evolutionary computation* 11(2):265–286.

Portelas, R.; Colas, C.; Hofmann, K.; and Oudeyer, P.-Y. 2020. Teacher algorithms for curriculum learning of deep rl in continuously parameterized environments. In *Conference on Robot Learning*, 835–853. PMLR.

Prideaux, D. 2003. Curriculum design. *Bmj* 326(7383):268–270.

Racaniere, S.; Lampinen, A.; Santoro, A.; Reichert, D.; Firoiu, V.; and Lillicrap, T. 2019. Automated curriculum generation through setter-solver interactions. In *International Conference on Learning Representations*.

Reehuis, E.; Olhofer, M.; Emmerich, M.; Sendhoff, B.; and Bäck, T. 2013. Novelty and interestingness measures for design-space exploration. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, 1541–1548.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17(1):67–99.

Romac, C.; Portelas, R.; Hofmann, K.; and Oudeyer, P.-Y. 2021. Teachmyagent: a benchmark for automatic curriculum learning in deep rl. In *International Conference on Machine Learning*, 9052–9063. PMLR.

Sakamoto, Y.; Ishiguro, M.; and Kitagawa, G. 1986. Akaike information criterion statistics. *Dordrecht, The Netherlands: D. Reidel* 81(10.5555):26853.

Schaal, S. 2006. Dynamic movement primitives-a framework for motor control in humans and humanoid robotics. In *Adaptive motion of animals and machines*. Springer. 261–280.

Schmidhuber, J. 2009. Simple algorithmic theory of subjective beauty, novelty, surprise, interestingness, attention, curiosity, creativity, art, science, music, jokes. *Journal of SICE* 48(1).

Sutton, R. S., and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.

Varshney, L. R.; Pinel, F.; Varshney, K. R.; Bhattacharjya, D.; Schörgendorfer, A.; and Chee, Y.-M. 2019. A big data approach to computational creativity: The curious case of chef watson. *IBM Journal of Research and Development* 63(1):7–1.

Walck, C. 2007. Hand-book on statistical distributions for experimentalists. *University of Stockholm* 10:112–113.

Wiggins, G. A. 2006. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19(7):449–458.

Wundt, W. M. 1874. *Grundzüge der physiologischen Psychologie*, volume 1. Wilhelm Engelmann.

Xu, Z., and Tewari, A. 2021. On the statistical benefits of curriculum learning. *arXiv preprint arXiv:2111.07126*.

Zhang, Y.; Abbeel, P.; and Pinto, L. 2020. Automatic curriculum learning through value disagreement. *Advances in Neural Information Processing Systems* 33.

Zhou, B.; Zeng, H.; Wang, F.; Li, Y.; and Tian, H. 2019. Efficient and robust reinforcement learning with uncertainty-based value expansion. *arXiv preprint arXiv:1912.05328*.

How to Report the Contributions of a CC System?

Anna Kantosalo, Simo Linkola, and Tomi Männistö
Department of Computer Science
University of Helsinki, Finland
anna.kantosalo / simo.linkola / tomi.mannisto @helsinki.fi

Abstract

We argue that the lack of well established reporting practices for applied Computational Creativity systems is hindering progress in the field. We consider that the current lack of reporting details – and variation in form and content – makes it difficult for third parties to reliably evaluate and compare systems based on publicly available information. This hinders forming an understanding of the similarities, differences and relative qualities of these systems. We propose a set of building blocks for robustly reporting the contributions of computationally creative systems to promote visibility and clarity in the field.

Introduction

The field of Computational Creativity (CC) is growing and reaching new levels of maturity. As the field attracts new audiences and new participants, it needs to make the research approachable and easy to understand through transparency. One of the key issues for transparency in applied creative systems is establishing field specific reporting practices. As the field matures, we have seen some gradual change towards better reporting practices. For example, Jordanous (2012b) has suggested practices for reporting CC evaluation. However, no comprehensive guide exists so far to support structuring CC reports for various audiences considering the basic elements of an applied system from a CC perspective. Therefore, we sketch out building blocks to support the transparent reporting of applied creative systems. These building blocks can be directly applied to support authoring specific sections of an applied CC paper.

Applied CC is set apart from theoretical CC, consisting of philosophy and methods of CC, by its focus on implementing systems that generate, evaluate or both generate and evaluate creative artefacts. The systems can be autonomous, interact with humans, or consist of several (autonomous) agents interacting with each other. They are often built to demonstrate a specific new CC method and increasingly deployed in real world contexts to aid real world creators. We outline reporting principles for such applied systems to improve communication within the field of CC and with the general public.

We argue that good reporting of applied creative systems should support *transparency* (see e.g. Fidler and Wilcox (2022) or Tearse, Wardrip-Fruin, and Mateas (2010)), which is a requirement for *reproducibility* (see e.g. Fidler and Wilcox (2022)) and allows for *system comparison*. As a stretch goal we consider that great reporting practices should also support communication to scientists, practitioners and the general public and relate new discoveries to previous progress in the field, following principles previously found useful in design science (Johannesson and Perjons 2014).

We propose three building blocks to support good reporting practices for applied creative systems. These building blocks can help authors to decide what to include in their applied CC research papers. Our aim is to supplement existing writing guides from related fields. Our building blocks are tailored to include aspects specific to creative systems, such as definitions for creativity. We next present our contributions and then discuss how they connect to general principles of good research and current practices in the field.

Building Blocks for Describing Computationally Creative Systems

We consider that at the heart of a successful applied CC paper is a robust description of the CC system and its contributions. We argue that the *description* of the system and its *evaluation* should go hand in hand with a *definition* of creativity fit for the *context* the system operates in. It is the mapping between this definition and the system description that allows the reader of an applied CC paper to contextualise the system and its contributions in the larger framework of CC research. We discuss these three parts in detail below.

Building Block 1: Definition of Creativity

A working definition of creativity allows the reader of the applied CC paper to situate the work within the larger scope of the CC field. A well chosen definition also allows readers from other, connected disciplines, as well as laypeople to understand how the applied research connects to our general understanding of creativity. In short, a well selected working definition for creativity manages the reader's expectations.

What sets a *working definition of creativity* apart from a *general definition of creativity* is that the definition does not need to be exhaustive: It can focus on a specific aspect of creativity, which is of interest to the researchers developing the applied system, or the domain the system operates in.

The best working definitions are short and refer to the larger body of literature on defining creativity (see e.g. Runco and Jaeger (2012)).

The authors of the applied CC paper should explicitly argue how the working definition of creativity connects to the *creativity goals* of the system. These are goals directly linked to the creativity of the system. If creativity is not the main goal, or the only goal of the system, authors should argue how the creativity goals significantly support the other goals of the system. For example the goal of a co-creative system may be to aid a user in a design task. This goal can be attained in many ways, but an important sub-goal, directly linked to creativity, could be the generation of valuable and novel design suggestions for the user. This way, in addition to setting expectations, motivating research, and connecting new research to existing research on CC, an explicit working definition of creativity promotes the transparent selection of suitable evaluation criteria for a CC system (Jordanous 2012b).

Building Block 2: System Description

A successful description of a CC system consists of several parts. The importance of each part depends on the *the scope of the system*, the stage of its *life-cycle* and *system goals*. Defining these explicitly is important to direct the readers' attention, manage expectations, limit scope, and set the context of the work.

The scope of the system should clarify if it is a full system, a part of a larger system or possibly a system embedded in a larger context or ecosystem of other systems. *The system life-cycle stage* should describe if the system is new, or a more established one, setting the expectations for the description and evaluations of the system. Finally, the *system goals* should connect to the chosen *creativity goals* and the *definition of creativity*.

The Generation-Evaluation Process. Typically a computationally creative system includes a part that generates creative outputs. The description of the *generator* should be detailed enough to enable the reproduction of a similar generator. The authors should at least answer the following questions: What kind of artefacts does the generator produce? What are the properties, and desired properties of the artefacts? What methods does the generator use to produce the artefacts? What kind of an architecture does the generator have and how does it connect to the rest of the system? Which data sets (or inspiring sets) does the generator use? If the system relies on a generative model requiring training, how was the model trained and what kind of parameters were used? If pre-trained models were used, what were they trained with and why are these models suitable for the creative purpose?

Correspondingly, many creative systems contain an internal evaluation component, or a component evaluating the creative contributions of other members of creative collectives. The *evaluator* should be documented with similar scrutiny to the generator.

If the generator and/or the evaluator are key contributions of the paper, the description of them must help the readers to understand how they work exactly. This requires comparing the generator/evaluator to existing generators/evaluators, which either produce or evaluate artefacts of the same kind or use similar processes in different domains, and explicitly pointing out the differences. If the generator or the evaluator consists of multiple parts, ablation studies are a good way to show how each of the subcomponents of the generator/evaluator affect the produced artefacts. This may require building mock or dummy implementations of each of the subcomponents. While ablation studies may seem like extra work, they tremendously support the transparency and comparison of the systems, and should be seriously considered in any system where the generator and/or the evaluator is part of the contributions.

Interfaces & Communication. For systems that interact either with humans or other systems, documenting the interaction *interfaces* is equally important. A short use case and/or a diagram illustrating how a human (or a machine) would interact with the system can be used to describe many aspects of an interface in an easy to understand manner. For visual interfaces this can be augmented with images and samples of other types of interface modalities can be included in external materials, such as video or audio. Whether a system interacts with a human or another system, it is also important to consider the following questions: Why does the system communicate with others? How does it happen? With whom? What kind of information is sent, and received? And finally, what triggers communication?

System History. Depending on the life cycle stage of the system some amount of the *history* of the system may be required for understanding it. History is especially important for studies building on existing systems: What version of the system is used? How does it differ from previous versions of the same system? In most cases it is good to explicitly answer the question: ”What is the new contribution this version of the system makes (also for creativity)?” For papers that primarily demonstrate improvements to existing systems, it is important to also document changes made to the algorithms and models used in detail. The reader should have a clear idea how the system components are changed compared to older versions and what the expected (or assessed) benefits of the changes are.

Ideally the history can also include core elements of the design process of the system: What important design decisions were made during the development of the system and how do they support the system goals and its creativity? A design decision can be for example what data set is used as an inspiring set for the system. It is important to document the expected benefits of the chosen approach with respect to the creativity goals of the system.

Finally an increasing number of systems learn and change during their life-cycle. These adaptive systems should describe also what changes during the run of the system, how the changes are triggered, and what contributes to them.

Building Block 3: Evaluation & Contributions

Evaluation in CC can refer to several different concepts: Internal evaluations conducted by the computationally creative system, or external evaluations aimed at summative or formative judgements of the quality and development areas of the system, possibly in a specific context. Similarly evaluation can be conducted by not only the system itself, but by system developers, or a third party, such as experts or laypeople. Full details on methods of evaluating applied CC systems is beyond the scope of this paper and there are several perspectives to evaluation that can be taken, including not only the evaluation of the creativity of the system, but also the fit of the system in the overall creative context it operates in. We refer the interested reader to Agres, Forth, and Wiggins (2016) or Jordanous (2012b) for more detail. Here we focus on evaluation as a relevant part of communicating the contributions of an applied CC system.

At minimum, documentation of an evaluation should explain what is evaluated, by whom, how, where and why. These questions help readers to assess if the evaluation of a system is robust, if it generalises to other audiences and contexts, possible sources of bias and if the evaluation is relevant. The documentation of the procedure also allows for reproducing the evaluation or conducting a similar evaluation on another system contributing to reproducibility and comparison of CC systems.

To be meaningful and relevant, the evaluation must be tied to the *creativity goals* of the system. As a core, extraordinary claims demand extraordinary evidence, therefore the chosen evaluation method and metrics should support the claims the authors of the system make about its creativity. This means the authors should document what metrics were used in the evaluation of the system and how do these link to the goals of the system and the chosen definition of creativity. So far there are very few established evaluation metrics presented in the field and some authors develop their own metrics or loan metrics from related fields. Echoing Jordanous, (2012b) it is important to establish why these metrics work in the chosen context so that the relevance of the evaluation can be assessed. Similarly, for an author to claim a system is creative, it is also important to document the self-evaluation metrics used by the system.

Discussion

We start with a brief discussion of the scientific objectives of the building blocks. We then discuss how the building blocks fit to the larger context of academic writing advice and connect with reporting practices from related fields.

We consider applied CC research as a discipline under the umbrella of Design Science. Similar to applied CC research, Design Science is a research paradigm that seeks explanations, predictions and descriptions for the current world, by actively trying to improve it through the creation of new systems (Johannesson and Perjons 2014, p.1).

Scientific Objectives for the Building Blocks

The purpose of our building blocks is to support three key ideas: *transparency, reproducibility, and comparing contributions within CC*. We consider that current weaknesses in reporting threaten these ideals and therefore hold back the progress of the field.

Transparency is a facet underlying the other two key ideas we wish to support. With transparency we refer to making information about the analysis and methods used accessible to the reader in a way that supports constructing an unbiased understanding of the applied CC system. The content of the blocks supports attaining this goal as the reader of a paper following the suggested block structure is more easily able to find the related information and make meaningful comparisons between systems.

Transparency is closely related to reproducibility in empirical science. Lack of transparency and completeness in method reporting (Fidler and Wilcox 2022) or datasets (Tearse, Wardrip-Fruin, and Mateas 2010) hinders reproducing previous experiments and the re-creation of systems. In addition, lack of transparency can render some CC evaluation methods useless, and impede with the independent evaluation of systems and research results.

For example Ritchie's (2007) criteria for evaluating creative outputs requires knowing the inspiring dataset used by the generator, as well as having access to a sufficient sample of results. If these are not stated, an independent evaluator cannot evaluate the applied CC system built by another, hindering for example, the use of the system as a baseline for future evaluations.

Similarly important for independent evaluation is to know the objectives of the research and the definition for the type of creativity the researchers are striving to implement with their system; In her seminal paper on standardised evaluation in CC Jordanous (2012b, p.1) argues for ”stating what it means for a particular computational system to be creative, deriving and performing tests based on these statements”. The lack of defining creativity makes it difficult especially for a layperson to evaluate creativity (Jordanous 2012b), which may limit the use of applied CC research results by general audiences. Therefore, announcing a working definition for creativity would improve both use and verification of results, but still many applied CC papers only make implicit assumptions about creativity.

Moreover, applied CC research seems to rarely record and publish negative results. Jordanous' evaluation of five CC presentations showed that developers typically focus on a few specific aspects of creativity, leaving multiple aspects impossible to review (Jordanous 2012a, pp.217-219). Only in one case of the five systems Jordanous' evaluated with her colleague was information sufficient to give a poor review of an aspect of creativity (Jordanous 2012a, pp.217). By pointing out their working definition on creativity, authors can communicate their focus to the external evaluator, as well as more reliably report also their negative findings on a specific aspect of creativity.

Reproducibility of experiments is a cornerstone of credible science. The so-called replication crisis brought the validity of results in medical, life and behavioral sciences into question in the 2010s (Fidler and Wilcox 2022). The definition of reproducibility varies between fields (Fidler and Wilcox 2022), here we refer to the ability to redo computations or whole experiments in principle and in practice, with the expectation of producing the same or sufficiently similar results. It can be further described as conceptual replications focused on verifying underlying hypotheses and direct replications aimed at controlling for samples, artifacts, fraud or generalization (Fidler and Wilcox 2022).

Similar to design science, replication in applied CC research can foster the accumulation and development of design theories and to encourage the reuse of designed systems and existing theories (Brendel et al. 2021). Currently the failure to reuse systems and connect studies to existing knowledge is limiting the contributions and effect of design science research (Brendel et al. 2021). We find this to be true for applied CC research as well: Especially the lack of robust documentation hinders progress and replication in the field, valuable knowledge lost, when specific systems loose their financial support and the systems and the related infrastructure is abandoned. It is of immediate concern that many of these tools cannot be reproduced as sufficient documentation of their development is not provided.

Replication studies in applied CC research are very scarce, and difficult to conduct as well. One of the few studies that could be considered a replication study in applied CC, is the re-creation of the Minstrel system reported by Tearse, Wardrip-Fruin, and Mateas (2010). This attempt to reconstruct a seminal system in computational story generation struggled with lack of original documentation, as for example the dataset used by Turner in developing the original system was undocumented. We argue that there are several other system, the recreation of which would be impossible, as we lack not only the data used in their creation, but sometimes also sufficient detail of the system architecture and implementation.

Finally, the lack of robust documentation hinders *comparing contributions made within CC*. This can mean the comparison of computationally creative systems overall, comparison of systems within the same creative subdomain, or even the comparison of a system with its earlier installations. The practical development of systems is driven especially by formative feedback (Jordanous 2012b). More documentation is required for formative evaluation tools such as SPECS (Jordanous 2012b) to be applicable to systems by outside evaluators. Alternatively, evaluations conducted by researchers themselves should be reported more openly and thoroughly. Similarly, for the purposes of scientific integrity, different editions of the same system should clearly document differences among the different editions of the system so that specific data can be connected with a specific implementation of the system creating a more robust system history benefiting practitioners in developing similar systems in the future.

The Building Blocks as Writing Advice

The building blocks suggested above could also be alternatively titled as a CC system documentation checklist, for they are largely based on the authors' experiences in participating in peer reviewing processes for papers describing CC systems. The critique presented most often seems to deal with establishing what it means for the system to be creative (Block 1), documenting the generation procedures in enough detail (Block 2), or showing a meaningful evaluation of the results (Block 3).

The blocks are also linked to the larger concept of academic writing advice. As we consider the field of applied CC inherently as a part of design science, the CC system including the generator naturally becomes one of the key items to document in research communications. Here we have only focused on aspects related to CC specifically. We therefore refer the interested reader to more specific advice on writing papers for design science (Johannesson and Perjons 2014, p.153-154).

We are also aware that as a multidisciplinary paradigm applied CC research has a lot to draw from related disciplines. We would for example argue that to a degree the adoption of neural nets in the generators offers great chances to draw from well established documentation practices in that specific area of artificial intelligence. Similarly in building interactive or co-creative computational systems, we have learned and adopted practices of evaluation with humans from interaction design. The purpose of this writing guide is therefore not to be definite, but we hope it works together with experiences from other disciplines to support a more robust reporting practice in applied CC research.

Conclusions and Future Research

While we do not particularly focus on evaluation here, it is clear that the diverse reporting practices contribute to the 'methodological malaise' in CC evaluation identified by Jordanous (2012b) and others. The lack of sufficient, accurate and accessible reporting of CC systems is contributing to a situation where reproduction of systems, and transparent evaluation by third parties, or the comparison of different systems or the different editions of the same system cannot be conducted. This hinders progress as we cannot leverage the full potential of applied CC research and build on the findings and work of others, establishing a robust, continuous base of evidence for improving machine creativity.

We have suggested three building blocks: a definition of creativity, description of the CC system and its evaluation to support applied CC researchers in communicating the contributions of their systems to different audiences. To further support transparency in CC research we encourage developing more formal languages for the description of CC systems in ways that can also be archived for future research. This could include experimenting with existing descriptive languages like UML, or ontologies such as OWL. We would also like to encourage authors to share implementations of applied CC systems. Good implementations could be gathered and made accessible online for example similar to the deep learning Model Zoo[1] project. In the future, we intend to conduct a literature review to further examine the weak points in the reporting practices of applied CC systems.

Author Contributions

The original idea for the paper came from the authors AK and SL. AK wrote the majority of the paper with SL. TM commented on the idea and the draft, with insights on documentation in software engineering and defining the scope of the work.

[1] https://modelzoo.co/ accessed 23rd of May 2022.

Acknowledgments

Funded by Academy of Finland (Grant #328729). We would like to thank our colleagues Dr. Anna Jordanous and Prof. Hannu Toivonen and the anonymous reviewers for their comments on the draft.

References

Agres, K.; Forth, J.; and Wiggins, G. A. 2016. Evaluation of musical creativity and musical metacreation systems. *Comput. Entertain.* 14(3).

Brendel, A. B.; Lembcke, T.-B.; Muntermann, J.; and Kolbe, L. M. 2021. Toward replication study types for design science research. *Journal of Information Technology* 36(3):198–215.

Fidler, F., and Wilcox, J. 2022. Reproducibility of Scientific Results. In Zalta, E. N., ed., *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Spring 2022 edition.

Johannesson, P., and Perjons, E. 2014. *An Introduction to Design Science*. Springer International Publishing, 1st ed. 2014. edition.

Jordanous, A. 2012a. *Evaluating computational creativity: a standardised procedure for evaluating creative systems and its application*. Ph.D. Dissertation, University of Sussex.

Jordanous, A. 2012b. A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. *Cognitive Computation* 4(3):246–279.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17:76–99.

Runco, M. A., and Jaeger, G. J. 2012. The standard definition of creativity. *Creativity Research Journal* 24(1):92–96.

Tearse, B.; Wardrip-Fruin, N.; and Mateas, M. 2010. Minstrel remixed: Procedurally generating stories. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* 6(1):192–197.

ICCC'22

June 27 — July 1
Bozen-Bolzano, Italy

13th International Conference
on Computational Creativity

ISBN 978-989-54160-4-2

9 789895 416042

Association for
Computational
Creativity

International
Conference
on Computational
Creativity